Anno II numero 12 - Ottobre/Novembre 1988 L. 9.000 - Frs. 13.50

COMMODORE professional

Noi 128 & 64

GRUPPO EDITORIALE JACKSON
AREA CONSUMER

La rivista specializzata per gli utenti Commodore

TOP GAME

Bob Morane

C 128

Data... Data...
Libro mastro per il bowling
Inventario della casa

C 64

Programmazione strutturata
Print loader
Un comodo Merge
Utility per il disk drive

Spedizione in abbonamento postale Gruppo III/70 Taxe percue (Tassa riscossa) Milano Ferr. Corr.

LE RIVISTE JACKSON PER IL TUO COMMODORE

IN EDICOLA O IN ABBONAMENTO: AMIGA MAGAZINE CON FLOPPY • AMIGA TRANSACTOR • COMMODORE PROFESSIONAL 64/128 CON FLOPPY E CASSETTA • SUPER COMMODORE 64/128 CON FLOPPY E CASSETTA

002P-10-88

PER TE
CHE HAI UN COMMODORE

È possibile imparare con facilità a lavorare, studiare e, perché no? Giocare. Come? Ma è semplice! Con VIDEO BASIC per capire tutti i segreti della programmazione e conoscere meglio il tuo computer; con il CORSO DI GRAFICA, per colorare di immagini il tuo video; con il CORSO DI SCACCHI per realizzare entusiasmanti partite; con il CORSO DI CHITARRA per imparare subito a suonare il tuo strumento preferito; con il CORSO DI MUSICA per acquisire facilmente tutta la materia e con la RACCOLTA DI 10 CASSETTE GIOCO per divertirti ininterrottamente, mesi e mesi.

Allora cosa aspetti? Approfitta di questa eccezionale e irripetibile offerta JACKSON

COMPILARE E SPEDIRE IN BUSTA CHIUSA A:
GRUPPO EDITORIALE JACKSON
VIA ROSELLINI 12 - 20124 MILANO

☐ RACCOLTA DI 10 CASSETTE GIOCO C64/C128/C64PC
L. 29.000 anziché L. 80.000

☐ VIDEOBASIC C64/C128/C64PC 20 lezioni + 20 cassette
L. 59.000 anziché L. 176.000

☐ CORSO DI GRAFICA C64/C128/C64PC 10 lezioni + 10 cassette
L. 44.900 anziché L. 96.000

☐ CORSO DI SCACCHI C64/C128/C64PC 10 lezioni + 10 cassette
L. 44.900 anziché L. 96.000

☐ CORSO DI CHITARRA C64/C128/C64PC 1 volume + 1 floppy
L. 9.900 anziché L. 15.000

CORSO DI MUSICA CON TASTIERA (15 lezioni + 15 cassette + 1 tastiera) ☐ Per Commodore 64 ☐ Per Commodore 64 PC
L. 54.900 anziché L. 156.000

Per un totale di L. _____
+ L.4.000 contributo fisso spese di spedizione

GRUPPO EDITORIALE JACKSON

MODALITÀ DI PAGAMENTO

☐ Allego assegno n. _____ di L. _____

della Banca _____

☐ Ho effettuato il pagamento di L. _____ a mezzo:
☐ vaglia postale ☐ vaglia telegrafico ☐ versamento sul c/c postale n. 11666203 intestato a Gruppo Editoriale Jackson SpA Milano e allego fotocopia della ricevuta

☐ Pagherò al postino l'importo di L. _____ al ricevimento

☐ Vi autorizzo ad addebitare l'importo di L. _____ sulla carta di credito:
☐ Visa ☐ American Express ☐ Carta SI ☐ Diners Club.

conto n. _____ data di scadenza _____

DATA _____ FIRMA _____

COGNOME _____ NOME _____

VIA E NUMERO _____

CAP _____ CITTÀ _____

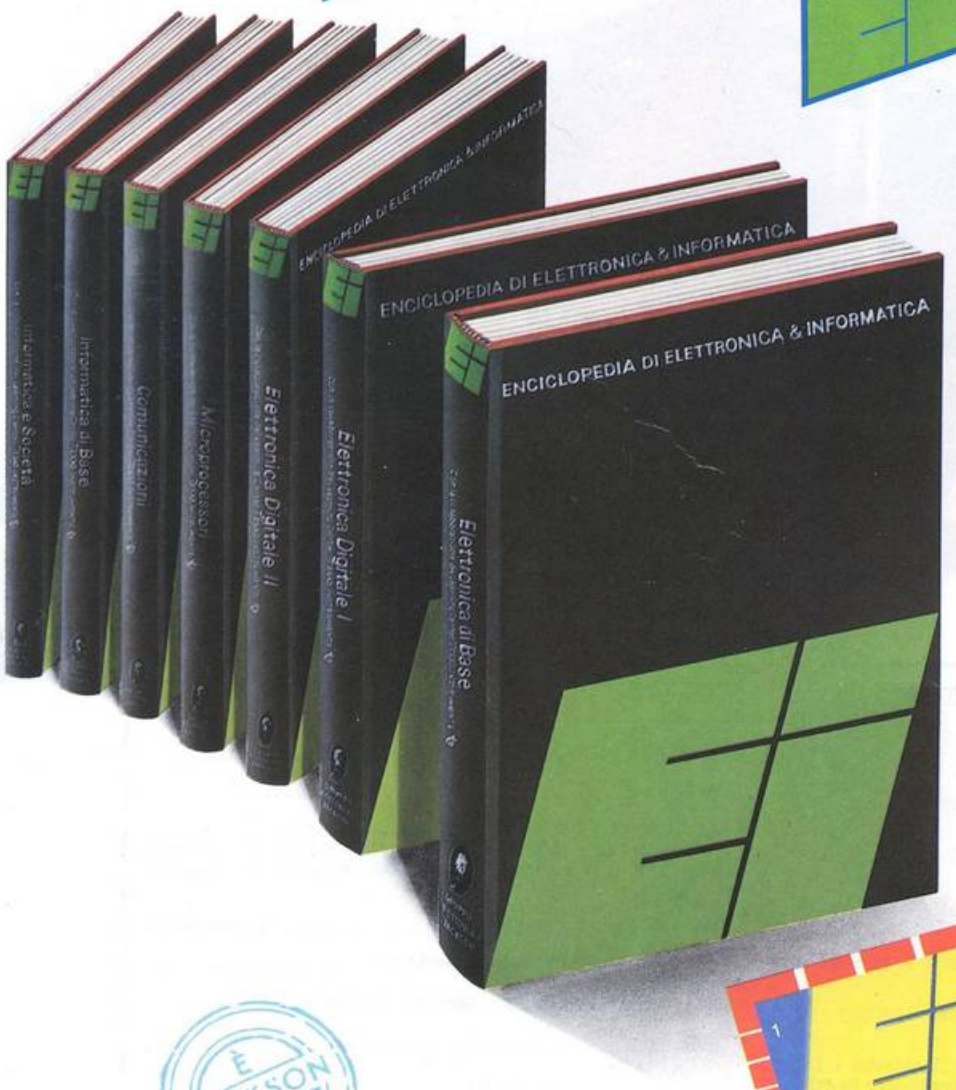
PROV. _____ TEL. (_____) _____

DAL 12 OTTOBRE IN EDICOLA

UN GRANDE RITORNO

Nuova edizione

ENCICLOPEDIA DI ELETTRONICA INFORMATICA E COMUNICAZIONI



Fascicolo dopo fascicolo, in sole 30 settimane, si completano i 7 grandi volumi di EI: l'Enciclopedia Jackson di Elettronica, Informatica e Comunicazioni.
EI è l'appuntamento settimanale in edicola con la tecnologia più avanzata; ogni fascicolo l'emozione di nuove scoperte; ogni pagina la certezza di vivere da protagonista la rivoluzione tecnologica in atto. EI, oggi in edicola, domani nella tua libreria.

GRUPPO EDITORIALE JACKSON
DIVISIONE GRANDI OPERE

Il 1° e il 2° fascicolo a sole L. 500

È JACKSON

LA TUA ENCICLOPEDIA

Omaggio a Marisa Bellisario

Con questo numero entriamo definitivamente in una nuova era della rivista Commodore Professional. Come già annunciato sullo scorso numero, la rivista è ora interamente dedicata alle piccole grandi macchine Commodore: il PC 128 e il C64. Per ben inaugurare questo aureo periodo - almeno lo speriamo - abbiamo infarcito la rivista di programmi succulentissimi: che non potranno deludere nemmeno i palati più fini. Per gli utenti del PC 128 presentiamo una serie di programmi utilissimi a tutti quei programmatori che trovano particolare piacere nel "giocare" con i numeri esadecimali del Linguaggio Macchina. Per loro, o meglio per i poveri mortali che poi devono ricopiare tali programmi, un set di programmi che vanno sotto il nome generico di Datamaker. Sempre per gli utenti C 128 un comodo programma di inventario, espressamente dedicato alla catalogazione degli oggetti che si possono trovare in casa. Questo è particolarmente utile a tutti coloro i quali hanno un'assicurazione contro i furti. Nella speranza che non debbano mai utilizzarlo per tale spiacevolissimo scopo. E nella speranza che non venga loro rubato anche il programma... Per chi usa il mitico C 64, un simpatico e didatticamente interessante programma di simulazione della programmazione strutturata in Pascal. Inoltre„programmi sulla gestione del drive, un particolare e sempre utilissimo programma di Merge, un ulteriore articolo sugli interrupt e molto altro ancora. Riprendendo il discorso iniziale, c'è da dire che le trasformazioni, speriamo in meglio, in meglio per chi ci legge ovviamente, non sono finite; ora voi vi aspettate che ve le elenchi, invece vi do appuntamento al prossimo numero per farvele toccare con mano. Sadico vero?! Arrivederci.

La Redazione

COMMODORE professional • 128 & 64

JACKSON

Anno II n. 12 Ottobre/Novembre 1988

DIRETTORE RESPONSABILE
Giampietro Zanga

DIREZIONE EDITORIALE
Daniele Comboni

REDAZIONE
Graphic & Comp. Gorizia

COORDINAMENTO REDAZIONALE
Simone Concina

ART DIRECTOR
Gianni Marega

COLLABORATORI
Furio Lusnig, Mr. Lambda, Stefano Sentieri, Alessandro Prandi, Primoz Beltram, Tomi Beltram, Avelino De Sabbata, Giorgio Dose, Donda Alberto

GRAFICA, IMPAGINAZIONE, COPERTINA
Graphic & Comp.

REVISIONE GRAFICA E IMPAGINAZIONE
Gianni De Tomasi

DIVISIONE PUBBLICITÀ
Via Pola, 9 - 20124 MILANO - Tel. 69.481
Telex 316213 REINAI - 333436 GEJ - ITI
OVERSEAS DEPARTMENT: Tel. 02/6948492
PUBBLICITÀ PER ROMA-LAZIO E CENTRO SUD
Via Lago di Tana, 16 - 00199 Roma
Tel. (06) 8380547 - Telefax (06) 8380637
00198 ROMA - Tel. (06) 8119803/4

FOTOCOMPOSIZIONE
FOTOFORMA - Via del Molino a Vento, 72
34137 TRIESTE

STAMPA
Centro Poligrafico Milano-Rozzano (MI)

DISTRIBUZIONE
Sodip - Via Zuretti, 25 - 20125 MILANO
Spedizione in abbonamento postale Gruppo III/70
Pubblicità inferiore al 70%

UFFICIO ABBONAMENTI
Tel. (02) 6122527-6187376
Prezzo della rivista L. 9.000 con cassetta;
L. 13.000 con disco
Abbonamento L. 59.500 (11 numeri con cassetta);
L. 85.000 (11 numeri con disco).
Abbonamento Estero rispettivamente
L. 119.000 e L. 190.000
mediante emissione di assegno bancario, vaglia o utilizzando il C/C postale numero 11666203
Per cambi di indirizzo, indicare, oltre al nuovo, anche l'indirizzo precedente.

GRUPPO EDITORIALE JACKSON
AREA CONSUMER

PUBLISHER
Filippo Canavese

DIREZIONE, REDAZIONE, AMMINISTRAZIONE
Via Rosellini, 12 - 20124 Milano
Tel. (02) 66.80.00.00-66.80.01.61-66.80.02.38
Telex 333436 GEJIT I

SEDE LEGALE
Via Pietro Mascagni, 14 - 20122 Milano

Il Gruppo Editoriale Jackson
è iscritto nel Registro nazionale della Stampa
al n. 117 vol. 2 - foglio 129 in data 17/8/1982

Autorizzazione del Tribunale di Milano n. 554
del 29.7.1987

C 128 & C 64

18 C 64 & programmazione strutturata

Nell'articolo verranno introdotte alcune nuove istruzioni, acquisite da un linguaggio pseudo Pascal, utilizzabili dagli utenti del C 64 che permetteranno loro di utilizzare direttamente le strutture

26 Utility per il disk drive

Le routine che presentiamo vogliono essere tali da completare la serie di utility per il drive, che sono state fino ad ora pubblicate. La prima fornisce ai file programmi l'opzione di auto-run, la seconda, invece, è utilizzabile a protezione di qualsiasi tipo di file

36 Print Loader

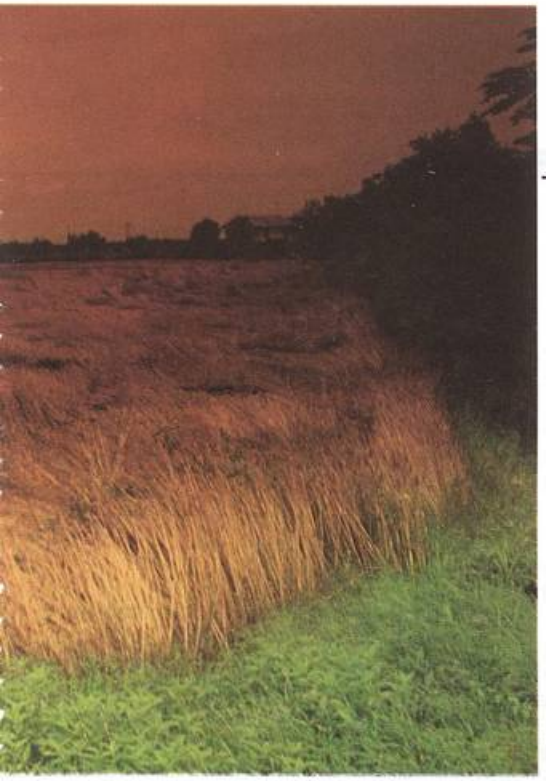
I programmi Basic spesso utilizzano brevi routine in Linguaggio Macchina per leggere il disco, ordinare i vettori o realizzare una certa funzione più velocemente e più facilmente di come si potrebbe fare con il Basic stesso

42 Data... Data...

Volendo continuare l'opera di normalizzazione intrapresa già da qualche numero, presenteremo ora una serie di programmi sia per C 128 che per C 64, che ci aiuteranno in futuro offrendoci la possibilità di proporvi listati più chiari di una sempre più semplice copiatura

68 Un comodo Merge

Molte tecniche di Merge sono state illustrate nell'immensa biblioteca dedicata al C 64, ma quella che vi faremo conoscere tra poco elimina un fastidioso procedimento altrimenti indispensabile nell'uso di questo atipico comando Basic



Sommario

ANNO II - NUMERO 12 - OTTOBRE/NOVEMBRE 1988

RUBRICHE

72 Libro mastro per il bowling

Migliorare i vostri punteggi e stabilire, serata dopo serata, dei nuovi record comporta anche un problema di come questi punteggi devono essere registrati; tale problema aumenterà la propria complessità qualora la gara per il miglior punteggio coinvolga diversi giocatori

78 Inventario della casa

Quando il fuoco o i ladri visitano la vostra casa, non c'è nulla al mondo che possa sollevarvi dalla sensazione di distruzione o violazione della vostra privacy; ma almeno, riuscite in parte a rassicurarvi pensando al fatto che la vostra assicurazione vi aiuterà a coprire parzialmente il danno subito. Ma voi siete in grado di ricordarvi tutto ciò che ora manca?

5 Editoriale

8 Lettere

10 Giochi

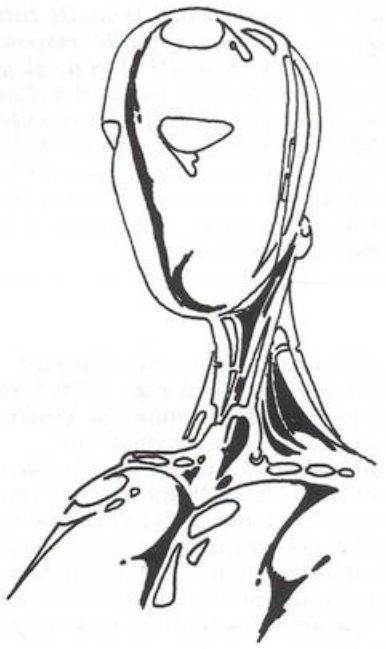
14 Immagini
Simone Concina

16 Miscellanea

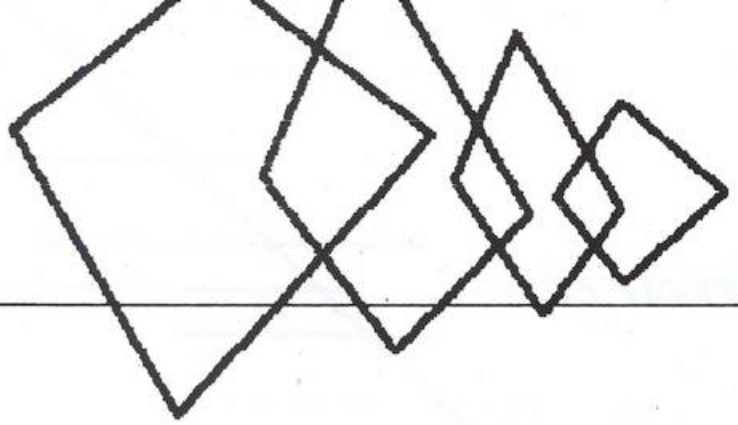
90 Top Game
Bob Morane

93 Anticipazioni

92 Errata corrige
Relativo a Fast Lister, pubblicato sul numero 10



Le assunzioni fotografiche delle opere in computer grafica sono state effettuate con attrezzature Polaroid Palette



Lettere

Spett.le redazione di Commodore Professional, innanzi tutto approfitto per complimentarvi con voi per la stupenda rivista.

Sono un soddisfatto possessore di un Amiga 1000 e vi scrivo perché, programmando in C, non riesco ad ottenere il carattere HOME (cursore in alto a sinistra dello schermo. Il carattere CLS lo ottengo con un DEFINE CLS "/033c".

Avrei anche bisogno di una funzione che mi permetta di caricare una font diversa da quella standard.

Sperando che continuiate sulla strada intrapresa e in una risposta a questi miei problemi vi saluto e vi auguro un meritato periodo di vacanze (forse è un po' presto! O no??).

Gianluca Terragno
Galatone (LE)

Innanzitutto ti ringraziamo dei complimenti. Speriamo ci perdonerai se cogliamo l'occasione di questa tua lettera per insistere sull'importanza di possedere i diversi manuali Amiga, anche se non ancora tradotti tutti in italiano. Per esempio, le risposte ai tuoi questiti e a molti altri analoghi si possono trovare nei capitoli 8 (Console Device) e 4 (Text), rispettivamente, del ROM Kernel Reference Manual: Libraries and Devices della Addison Wesley. E ora sapendoti in buone mani possiamo anche pensare alle vacanze... Ma ormai alle prossime!

Cari amici, siamo due studenti, nonché possessori di Amiga 1000 (nonché abbonati) e gradiremmo un Vostro aiuto di ordine 'programmatorio'.

Abbiamo infatti digitato il vostro programma in Basic 'Superfici 3D', pubblicato sul numero 1 della Vostra rivista Commodore Professional, non paghi di ciò abbiamo inoltre creato una Maschera di input che ci consente di modificare i parametri senza dover richiamare il listato.

Inoltre abbiamo aggiunto il comando CHAIN MERGE per poter cambiare le funzioni introducendole semplicemente da tastiera, aprendo un RAM-FILE mediante il quale viene poi sostituita automaticamente la linea 90 (che nel nostro caso contiene la funzione).

Al fine (AHINOI !) di ottenere un programma più veloce abbiamo cercato di compilarlo con l'ultima versione del A.B.C., ma come potete ben immaginare, il comando CHAIN MERGE sostituisce la linea Basic, e non già compilata ed il compilatore non ha (logicamente) riconosciuto il suddetto comando.

Ci vediamo pertanto costretti a chiedere un vostro aiuto per aggirare l'ostacolo, e speriamo vivamente che non ci lascerete nei guai.

Potreste dunque essere tanto gentili da pubblicare i passaggi che occorrono per poter compilare il programma in modo tale da poter sostituire la funzione con una semplice richiesta di input, e non dovendo per forza mantenere il listato Basic?

Confidando nella vostra cortesia cogliamo l'occasione per rinnovarVi i nostri complimenti per il livello veramente notevole della Vostra rivista.

Antonio Carai
Giovanni Tamponi
Cagliari

Rispondiamo volentieri alla vostra lettera poiché ci permette di affrontare un argomento di notevole interesse: l'introduzione di una funzione in un linguaggio compilato. Il problema, è però di soluzione tutt'altro che immediata, infatti per sua stessa natura un programma compilato non possiede quelle doti di interattività proprie di uno interpretato. Una soluzione parziale è quella di inserire più funzioni nel sorgente e quindi richiamarle all'occorrenza. La soluzione generale è quella di crearsi un vero e proprio interprete nel proprio programma: in ingresso preleverete allora la funzione come stringa (notate bene) che verrà quindi elaborata permettendo così la valutazione della funzione che rappresenta.

Purtroppo una tale tecnica richiede la realizzazione di un sottoprogramma di complessità tali da non permetterci una sua descrizione in questa sede. Naturalmente questo non significa arrangiatevi: cercheremo quanto prima di sviluppare l'argomento in un prossimo numero di Amiga Magazine.

Invitiamo comunque i lettori interessati a suggerire soluzioni diverse, saremo ben lieti di accoglierle sulla nostra rivista.

Spett.le Redazione, ho tre magnifici computer (o sono 4?), C-16, C-128 e Amiga 500. Assieme a due amici cominciammo a utilizzare il Basic 3.5 poi con il C-128 il Basic 7.0, ora che abbiamo l'Amiga uno di noi ha venduto il C-128 e l'Amiga non lo interessa, l'altro ci gioca e il mio Amiga viene usato dalle mie figliole (con programmi di grafica o di musica come il SONIX).

Perché i tre Basicomani non discutono più le loro macchiavelliche subroutine su chilometrici listati di carta rigata?

Perché AMIGABASIC è troppo complesso e l'editing dei programmi non è più un divertimento.

Lo testimonia un articolo apparso su Commodore Professional del Marzo 1988 (pg. 15) in cui risulta evidente che tra SCREENS, WINDOWS, GADGETS e MENU non si arriva mai al DUNQUE!

Nel tentativo di ristabilire le riunioni dei Basicomani ho mostrato ai miei amici ABASIC by METACOMCO V1.0. Con i suoi listati numerati ed i programmi su carta leggibili nelle loro varie gosub ecc., ma niente da fare i difetti sono innumerevoli così abbiamo lasciato il GURU alle sue meditazioni e il mio C-128 è diventato più prezioso che mai.

È evidente che per ristabilire un contatto col crescente panorama di Amiga tra gli amici, ho bisogno di un Basic che abbia almeno le caratteristiche del Basic 7.0, tra le quali irrinunciabili sono:

Numeri automatici di linea + renumber; chiara visibilità dei caratteri su 25×40 righe; finestra standard 320×200; cursor flashing; funzioni logaritmiche, trigonometriche, esponenziali e pi greco; correzione dei listati passandoci sopra con il cursore, grafica, sprite, ecc.

Vi ringrazio anticipatamente se saprete indicarmi la reperibilità di un tale programma.

Spero solo che per quanto costoso sia completo e affidabile, sul quale si possa ristabilire un lungo e interessante legame di collaborazione.

Armando Pellizzaro
Vicenza

Beh, che dire: l'AmigaBasic un passo indietro rispetto al Basic 7.0, sinceramente non ci pare. Non possiamo però non condividere la sua irritazione nei confronti dell'editor del Basic di Amiga: la lentezza nello scrolling o la fiscalità con cui segnala anche il più misero dei "sintax error" non sono davvero piacevoli.

Per quanto riguarda il resto non sappiamo davvero come rispondereLe: ci tiene tanto alle linee numerate? L'AmigaBasic non le vieta mica! Però non Le pare che GOSUB FaiQualcosa sia più significativo di GOSUB 12345? Cursore lampeggiante e schermo a 40 colonne...?

Indubbiamente l'Amiga è una macchina complessa, leggendo la sua lettera ci rendiamo però conto che non è la complessità che la spaventa, e allora perché non dedica un po' di tempo a scoprire le potenzialità di AmigaBasic; siamo sinceramente convinti che un po' di impegno da parte sua verrà generosamente premiato dai risultati che lei stesso otterrà.

Sono studente in ing. elettronica a Parma e frequento il primo anno, ultimamente mi sono interessato alla vostra rivista Commodore Professional che è arrivata al numero sei, ma in edicola sono riuscito a reperire tutti i numeri meno il primo, così ho deciso di scrivere la presente per chiedervi di spedirmi, se vi è possibile, il numero mancante.

Inoltre gradirei sapere, visto che girano molte voci e ritengo voi i più competenti per darmi tali risposte, se oltre all'Amiga 500 e all'Amiga 2000, è in vista una nuova macchina Commodore con nuovi chip grafici più potenti (1 Mb di chip RAM di Fat Agnus si dice), un nuovo Amiga DOS 1.3 (così sull'Amiga 500 e 2000 si dovrà cambiare ROM!), nuovi processori tipo 68020 o '030 (ultimo uscito dalla Motorola) e coprocessori matematici tipo 68882 in grado di aumentare fino a ottanta volte la velocità standard.

Fiducioso in una vostra sollecita risposta vi porgo i miei più distinti saluti, e... complimenti per le vostre pubblicazioni (come Automazione, ecc...).

Fabrizio Contini
Cast. Sotto (RE)

Forse il numero mancante le sarà già giunto. Come lei saprà di certo la Commodore Italiana non brilla certo per la velocità con cui presenta al mercato italiano le novità sia hardware che software. Preferiamo però non alimentare leggende su eventuali nuovi chip grafici, che avrebbero, secondo le voci, capacità quasi divine. (Ne riparleremo, non dubiti, appena li avremo sotto mano).

Se permette un consiglio (ci creda il nostro non è un interesse puramente venale) segua la nuova rivista Amiga Magazine, che essendo interamente dedicata ad Amiga meglio potrà rispondere alle sue curiosità.

La redazione di

Commodore Professional
cerca collaboratori esterni.

Si richiedono,
oltre a una buona conoscenza
di almeno una delle macchine
Commodore (C64, C128, Amiga),
la conoscenza della lingua inglese,
dei linguaggi informatici
e una buona capacità di redarre
testi in italiano.
Chi volesse collaborare
con la nostra redazione è pertanto
pregato di scrivere a:

Redazione
Commodore Professional
GRUPPO EDITORIALE JACKSON
via Rosellini, 12 - 20124 MILANO

oppure telefonare
al numero 0481-83927
dopo le ore 16.00.

Street sports baseball

Il giocare a baseball è quasi un diritto sancito dalla costituzione per i giovani che crescono all'ombra dello stendardo a stelle e strisce. Anche se il divertimento è identico in qualsiasi stato dell'Unione in cui venga praticato questo gioco, il terreno su cui viene svolto a volte è di natura molto diversa. Si può passare da appezzamenti di terra in cui i prati sono ben curati, alle periferie delle grandi città o addirittura sulle strade di qualche paesino sperduto. Lo spirito competitivo rimane comunque sempre acceso ed ogni partita viene disputata con il massimo impegno. Ed è questo spirito di continua competizione che la Epyx ha cercato di riproporre nel produrre questo nuovo game per il Commodore 64. Il gioco può essere proposto a giocatori di qualsiasi età e abilità.

La prima opzione del gioco vi permette di scegliere tra l'effettuare una partita su un prato o in un parcheggio. L'unica differenza tra i due luoghi è il tipo di ostacoli che potrete incontrare sui due campi. Dovrete cercare di intercettare delle palle volanti e vi assicuriamo che non è un compito facile quando ci sono ostacoli come uccelli, pozzanghere o cespugli di diversa natura. Naturalmente per rendere il gioco più difficile, la posizione e il tipo degli ostacoli varia da partita a partita facendovi perdere spesso delle palle che si stavano giusto infilando nel vostro guantone.

C'è la possibilità di giocare o contro il computer o contro un altro giocatore. Le squadre possono essere scelte in modo casuale dal computer oppure è possibile richiamare dalla memoria dei gruppi precedentemente registrati su dischetto. Se selezionate una nuova squadra avrete l'opportunità di conoscere i punti di forza e le rispettive debolezze di ciascun giocatore.

È interessante il modo in cui il gioco fornisce per ciascun giocatore una precisa identità individuale in termini di abilità competitive. Un altro punto per cui il gioco è simile alle situazioni di vita reale è il fatto che alcuni giocatori sono migliori di altri e che comunque ognuno di essi contribuisce a formare la fisionomia della squadra. Ognuno ha un nome ed una propria personalità, e sono incluse nel gioco anche delle figure femminili. Ma attenzione! alcune di queste sono veramente abili e vi daranno del filo da torcere.

Per determinare quale dei due gruppi batte per primo, verrà utilizzata una monetina. Ogni gruppo è formato da otto giocatori e il computer gestisce il catcher per entrambe le squadre e il pitcher si occupa della casa base dopo che la palla è stata colpita. Poi ogni giocatore decide quale sarà la sua posizione sul campo da gioco; ed è a questo punto che dovrete dimostrare la vostra abilità analizzando le capacità di ciascun giocatore e posizionandolo nel posto che ritenete più opportuno.

Sulla parte superiore dello schermo comparirà un tabellone per l'assegnazione dei punteggi e due diverse finestre visualizzeranno il campo da differenti angolazioni. La finestra più grande vi propone la porzione del campo in cui si svolge l'azione (il punto di vista si trova esattamente dietro la base) e cioè dove si trova attualmente la palla. La finestra più piccola propone una vista dall'alto del campo di gioco e vi sarà estremamente utile per rintracciare le traiettorie delle palle volanti, della posizione dei giocatori, ecc.

Le palle possono essere lanciate nei seguenti modi: lente, veloci, con effetto verso destra o verso sinistra e la finestra maggiore vi permetterà di lanciare la palla nel modo migliore. Vi divertirete a far impazzire i battitori con rapidi cambiamenti di velocità impressi alla palla o imprimendo loro degli effetti perversi. Ricordatevi però che le palle a cui viene impresso un effetto troppo marcato rischiano di colpire il battitore permettendogli così di camminare verso la casa base.

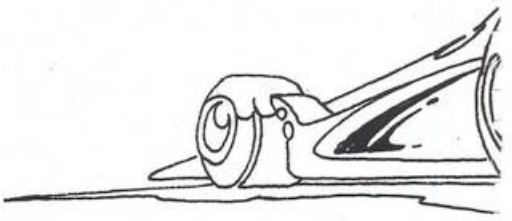
Analizzando il gioco dal punto di vista del battitore possiamo dire che colpire la palla non è difficile, basterà concentrarsi un po'. Vanno comunque considerate le differenze indiividuali di ciascun giocatore dal momento che ci sarà qualcuno più rapido nel colpire la palla e qualcuno invece no! Se manterrete sempre presente questo consiglio non incontrerete alcuna difficoltà.

Dopo aver colpito la palla, la finestra principale visualizzerà la porzione di campo più vicina alla palla in quel momento. L'altra finestra visualizzerà invece l'intero campo, il tracciato percorso dalla palla e la posizione dei giocatori. Immediatamente prima che la palla compaia sullo schermo un messaggio vi informerà se la traiettoria della palla è alta o bassa. Le palle volanti formano un'ombra sul campo ed è quella che dovrete seguire per avere la certezza di intercettarla quasi ogni volta.

Prendere la palla in modo corretto dipende però da due fattori: l'abilità del giocatore e gli ostacoli presenti sul campo. Scivolare nelle pozzanghere o incappare nei cespugli vi farà perdere tempo e la coordinazione necessaria per eseguire in modo corretto l'azione. Un giocatore che ha le mani di 'pastafrolla' rischia di vedersi sfuggire la palla dalle mani dopo aver effettuato una splendida presa oppure ancora di vedersi passare la palla in mezzo alle gambe.

Per passare la palla da una base all'altra non dovrete far altro che che puntare il vostro joystick su una delle quattro basi e premere poi il pulsante di fuoco. Il giocatore si sposterà automaticamente verso la base indicata recuperando la palla.

La grafica è discreta e i colori sono utilizzati in maniera ottimale. Per quanto riguarda l'animazione possiamo solo

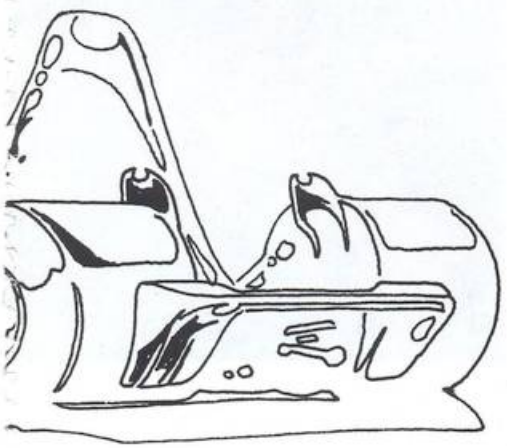


affermare che è veramente ben fatta. Determinante è l'apporto degli effetti sonori: il rumore della palla quando viene colpita dalla mazza o il suono occasionale del clacson di un'auto di passaggio tanto per ricordarne alcuni. Il gioco è ben fatto ed è adatto a giocatori di qualsiasi età, non è troppo impegnativo ed è divertente.

Aliens

Questo game prodotto dalla Activision, propone una collezione di minigame per il Commodore 64. Per la precisione i giochi contenuti sono sei e riproducono parti dell'omonimo film. I due personaggi principali vi sono inclusi, e voi dovrete determinare le azioni di Ripley in varie parti del gioco.

La prima parte del game consiste nel cercare di pilotare la navicella durante la discesa sul pianeta. Questa sezione del gioco non è una vera e propria simulazione di volo nella quale ci si deve cimentare con complicati controlli e perciò risulterà semplice e divertente. Una volta atterrati, verranno inviati in missione quattro uomini dell'equipaggio alla ricerca dell'APB. A questo punto del gioco compaiono gli alieni ed in questa seconda parte sarete impegnati in una ritirata strategica che sarà resa più difficile dalla preponderanza numerica del nemico che vi terrà inchiodati sulle vostre posizioni.



Raggiunta la salvezza verrete spostati in un'altra area del complesso APC. Ricordatevi che l'Armored Personnel Carrier è posto sempre nello stesso angolo della palazzina. Se la vostra squadra verrà circondata dovrete cercare di difendere la posizione finché uno dei vostri uomini non riesca ad attraversare la linea nemica e a scappare attraverso una porta d'acciaio. Questa parte del gioco vi indurrà ad analizzare molto velocemente la situazione dal momento che dovrete attraversare otto linee nemiche di alieni.

Lasciata alle spalle questa situazione per niente piacevole, Newt suggerirà che è molto più sicuro scappare attraverso i condotti d'areazione. In questo gioco nel quale voi e gli altri superstiti dovrete cercare di attraversare il labirinto dei condotti d'areazione cercando di sfuggire alla cattura altrimenti vi attenderà una morte lenta e per questo ancor più dolorosa. Ogni uomo è conscio del pericolo che sta correndo ed è pronto a sacrificare la propria vita per salvare il resto del gruppo, facendo detonare la propria bomba quando il gruppo è in pericolo. Purtroppo durante questa fase delle operazioni Newt cadrà nelle mani del nemico.

Anche se il reattore nucleare esploderà entro 17 minuti, Ripley metterà in pericolo la propria vita per cercare di salvare il suo giovane amico. Ed è in questa fase di salvataggio che si dovrà fare ancora più attenzione dal momento che i pericoli saranno numerosi. Per ritrovare il vostro amico sarete dotati di un detector per l'individuazione delle persone e Ripley cercherà di lasciare una traccia del suo passaggio. Anche se siete alla ricerca dell'orrendo alieno cercate di mantenere parecchie munizioni per quando incontrerete la regina nel nido degli alieni. Vi suggeriamo di sparare sul nido, in questo modo la regina cercherà di proteggere il nido e voi sarete liberi di procedere verso l'ascensore.

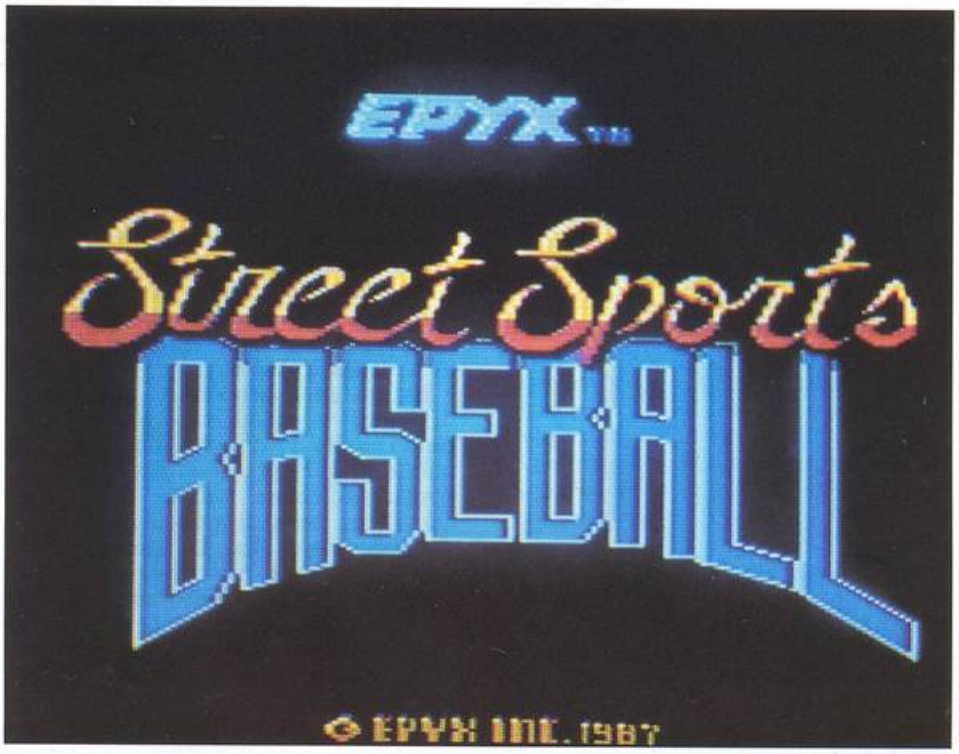
Nell'ultima parte del gioco avrete come avversaria la regina degli alieni. Durante questa mitica battaglia Ripley guiderà una macchina che la aiuterà a sconfiggere la regina. Cercate di vincere l'alieno e di espellerlo dalla navicella attraverso il condotto d'areazione, dal momento che sarà questa l'ultima fatica a cui sarete chiamati. La strategia vincente durante questo confronto è quella di continuare a colpire la regina con la vostra macchina in modo da indebolirla. Solo al momento opportuno cercate di catturare l'alieno e di espellerlo dall'astronave

La grafica è ben curata e l'alieno è disegnato in modo convincente, valutando anche il fatto che sono principalmente delle figure nere con alcuni particolari di colore bianco. Ricordatevi che alcuni alieni si muovono molto più velocemente di altri. La stanza operativa è altrettanto ben disegnata e così pure il divampare delle fiamme. Alcune schermate però sarebbero risultate più credibili se fossero stati introdotti maggiori dettagli. Ci sono alcune somiglianze con un altro gioco della Activision e precisamente con Masters of the Lamps nel quale il protagonista principale deve guidare un tappeto volante attraverso una serie di anelli mistici.

Il programma utilizza la colonna sonora del film Alien oltre a un consistente numero di effetti sonori. Nel progredire della storia udrete colpi d'arma da fuoco, lo scalpiccio di passi attraverso i vari tunnel, strani rumori emessi dalle creature aliene, ecc. Ogni gioco è fornito di un proprio numero di effetti speciali per sostenere le varie scene rappresentate. Il gioco ha perciò varie ragioni perché lo si possa definire unico nel suo genere. In primo luogo ricrea le scene cruciali dell'omonimo film e vi sfida a vestire i panni di uno dei personaggi coinvolti nella storia. Vi invita all'osservazione e a prendere delle decisioni rapide. Della buona musica, della grafica a tratti ottima fanno da supporto ai vari schemi. Si è ottenuto in questo modo un risultato finale brillante che vi terrà inchiodati al vostro joystick in una guerra privata contro una creatura estremamente intelligente che proviene da un altro mondo.

Cauldron

Avvicinatevi giovani giocatori dal momento che stiamo per servirvi un semplice ma fantastico piatto servito



dai disegnatori della Broderbund per coloro che sono in possesso di un Commodore 64. La ricetta consiste nel prendere una parte di adventure e due parti di arcade, mescolarle lentamente assieme e quando il tutto si è perfettamente amalgamato otterremo un fresco piatto che intratterrà anche i più reticenti.

In Inghilterra, dove è stato già introdotto, ha raggiunto in brevissimo tempo il vertice della classifica delle vendite ed ora che è approdato lungo le nostre spiagge, è facile prevedere che venga nuovamente baciato dal successo.

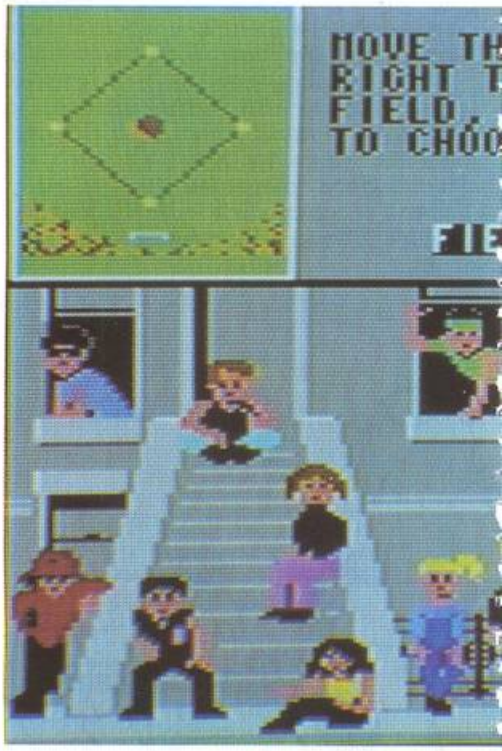
Cauldron però non è un gioco singolo ma ne racchiude due. Entrambi i giochi, o per essere più precisi le sfide, sono strutturati sulla rivalità esistente tra il re del Pump e la regina delle streghe. Questi due mortali nemici si sfidano in una lotta eterna che li incatena uno all'altra sulle terre a loro sottomesse.

In un primo momento assumerete il ruolo della regina delle streghe, ma non rallegratevi! la regina ha appena subito un incantesimo. Si dice che il suo più acerrimo nemico le abbia sottratto con l'inganno la sua preziosa scopa d'oro e che la tenga ben protetta in una segreta all'interno dei sotterranei del suo castello. Il vostro tesoro è sorvegliato da legioni di guardie ed evitarle non è un compito che si possa definire semplice. Non disperatevi, la vostra magia vi verrà in aiuto; dovrete cercare nei vostri libri una pozione, determinarne l'esatta mistura in modo da rendere inoffensivo il vostro nemico e i sui seguaci. Il gioco perciò si sviluppa in una caccia all'interno di tetri paesaggi nei quali cercherete di rintracciare degli ingredienti peculiari che una volta raccolti dovranno essere mescolati nel calderone per preparare la pozione magica.

Per rintracciare il succo di lucertola, l'occhio di un licantropo e gli altri deliziosi ingredienti necessari per la mistura dovrete attraversare foreste, mari e sorvolare montagne facendo attenzione alla moltitudine di esseri immondi che cercheranno di sconfiggervi con il loro tocco magico. Il lavoro è decisamente difficile ma alla fine otterrete il risultato meritato.

Il logico proseguimento di questa storia lo possiamo rintracciare nel Pumpkin Strikes Back che ci propone ciò che accadrà alcuni anni più tardi. Molte volte nelle leggende è possibile distinguere una certa specularità nella narrazione della storia. La strega dopo essere riuscita a rintracciare la sua scopa dorata ritorna ad essere la padrona di quelle lande incantate ed ora il suo unico scopo è quello di vendicarsi eliminando e terrorizzando tutti i suoi oppositori. Ora il vostro ruolo vi permetterà di calarvi nelle vesti del suo acerrimo nemico: il PumpKing. La missione di quest'ultimo sopravvissuto consiste nello sconfiggere la regina delle streghe e ristabilire il proprio potere.

Anche in questa avventura il vostro attacco potrà partire solamente dopo che avrete rintracciato tutti gli ingredienti necessari per preparare la pozione che dovrà essere introdotta nel calderone magico. La caccia agli ingredienti si svolgerà all'interno del castello del nemico. In questa cittadella fortificata risiede anche la strega, e voi dovrete cercare i vari elementi all'interno delle segrete, delle umide celle,



lungo gli oscuri corridoi e dietro porte con enormi borchie facendo attenzione ad evitare le guardie. Per proseguire nella vostra caccia oltre all'abilità avrete bisogno anche di una buona dose di fortuna.

Questi due arcade, soddisferanno anche gli smanettatori più esperti dato che per sventare gli attacchi si deve essere molto veloci nell'uso del joystick. Nel primo gioco il joystick che controlla la strega vi permetterà di farla camminare, saltare o farla volare nel cielo in sella al manico della sua scopa. Quest'ultimo mezzo di trasporto sarà quello più utilizzato e vi forniamo perciò alcune regole per utilizzare questo piccolo gioiello volante. Anche le scope volanti soggiacciono alle più elementari leggi della fisica quali il principio d'inerzia e dovrete perciò decelerare e fermarvi prima di poter invertire la marcia. Il raccogliere gli oggetti o l'atterrare richiedono un po' più di pazienza e di abilità.

In Cauldron II, Pumpking e i suoi accoliti sono privi di arti e l'unico modo che hanno a loro disposizione per spo-

starsi e quello di rimbalzare; quest'esercizio vi porterà in breve sull'orlo di una crisi nervosa. Dovrete cercare di evitare le varie trappole disseminate all'interno del castello fino a raggiungere il quartier generale della regina delle streghe. Il controllo del personaggio all'inizio è difficile ma dopo un po' vi abituerete a questa inusuale andatura.

Gli ambienti del gioco sono sviluppati con la tecnica del fumetto nel quale il reale ed il surreale sono magistralmente mescolati creando così un'atmosfera magica e surrealistica. L'animazione è piacevole e gli autori hanno tenuto conto anche dei dettagli: ad esempio gli abiti della strega sono mossi dal vento quando attraversa il cielo illuminato dalla luna o le bellissime bolle infuocate sulle superfici delle polle vulcaniche.

Il gioco non è molto semplice e i novelli giocatori vengono eliminati rapidamente, solmente coloro che si sono cimentati con gli arcade da lungo tempo hanno delle buone chanche per poter proseguire il gioco. Si sente la mancanza di una serie di mappe che illustrino questo mondo magico o almeno una serie di indicazioni sull'ubicazione dei vari ingredienti necessari per la costituzione della pozione magica. Il gioco perciò non è adatto ai più giovani o per coloro che vengono pervasi da un senso di frustrazione ai primi insuccessi.

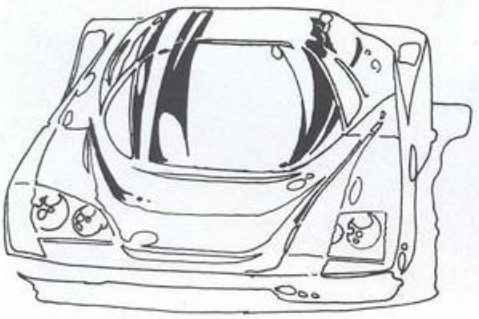
Nelle prime partite è bene scordarsi i punteggi ma invece si cerchi di allenarsi al controllo dei movimenti della scopa volante. Durante il percorso si cerchi di volare il più in alto possibile dal momento che i vari animali si muovono molto lentamente, in questo modo potrete proseguire più velocemente risparmiando tempo, energia e vite. Se invece siete all'interno dei corridoi del castello dovrete cercare di controllare perfettamente il vostro movimento (il rimbalzare), dal momento che il soffitto di questo corridoio è una trappola e perciò non va sfondato.

I disegnatori, mossi a compassione, hanno introdotto un'opzione di aiuto per poter uscire da eventuali trappole. Basterà premere il tasto 'R' per liberarvi, quest'azione vi costerà comunque una vita.

Immagini

Da questo numero Commodore Professional assume una nuova veste e ,con la rivista, anche la rubrica VERNISSAGE cambia: anzi scompare lasciando il suo posto alla rubrica "IMMAGINI".

Immagini, come lo dice stesso nome, è uno spazio dedicato all'immagine;



all'immagine, sia essa creata , rubata, vissuta, e in fine regalata a chi sa coglierne la grandezza.

Fin dal primo numero siamo partiti con l'idea, molto chiara che, il lettore, sfogliando questa rivista, non dovesse imbattersi in immagini fin troppo ricorrenti in pubblicazioni analoghe, quali tastiere, monitor ed hardware vari, bensì trovare nella rubrica "VERNISSAGE", un intermezzo a sè stante, più strettamente "figurativo" ed "artistico".

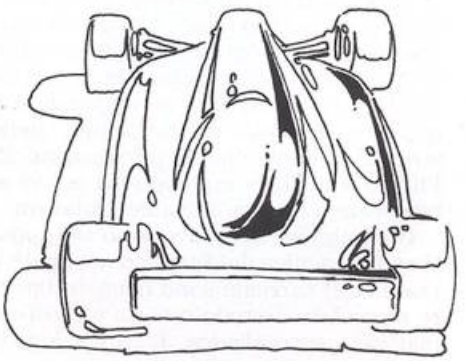
Perciò, nello spazio di "VERNISSAGE", abbiamo pubblicato numerosi prodotti artistici di nostri lettori, diversi per stile, rappresentazioni ed idee, ma tutti accomunati dallo stesso intento, ovvero lo sfruttamento delle innumerevoli possibilità grafiche del computer "AMIGA" Commodore.

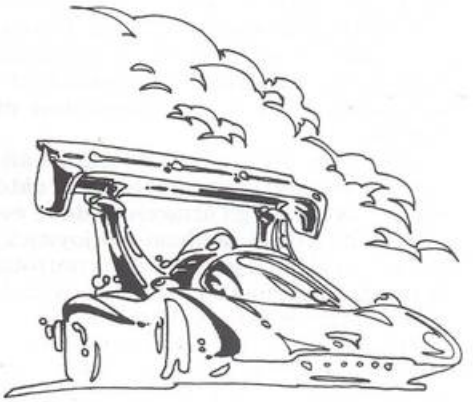
Da questo numero di "Commodore Professional", "AMIGA" se ne và, ha trovato uno spazio esclusivo in una rivista tutta sua: "AMIGA MAGAZINE". Termina perciò anche "VERNISSAGE", così indissolubilmente legato al computer in questione, lasciando posto ad "IMMAGINI".

"IMMAGINI" accoglierà qualsiasi tipo di espressione artistica, anche totalmente estranea al computer, dalla fotografia, alla pittura, e in generale quindi alla grafica, nel senso più ampio della parola stessa.

In questa prima puntata della rubrica, pubblichiamo un'interessante sequenza di immagini fotografiche, realizzata dal coordinatore redazionale della nostra rivista: Simone Concina.

Si tratta di paesaggi, scorci e vedute cittadine, catturate nei pochi momenti di tempo libero, rubati ai suoi impegni



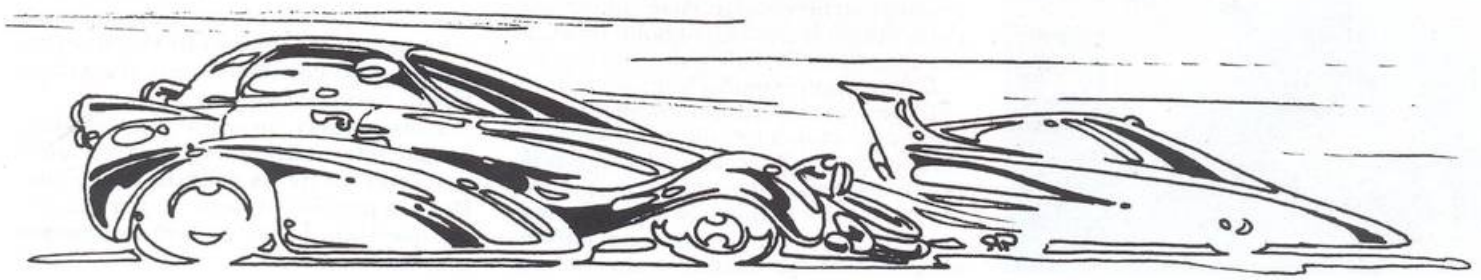


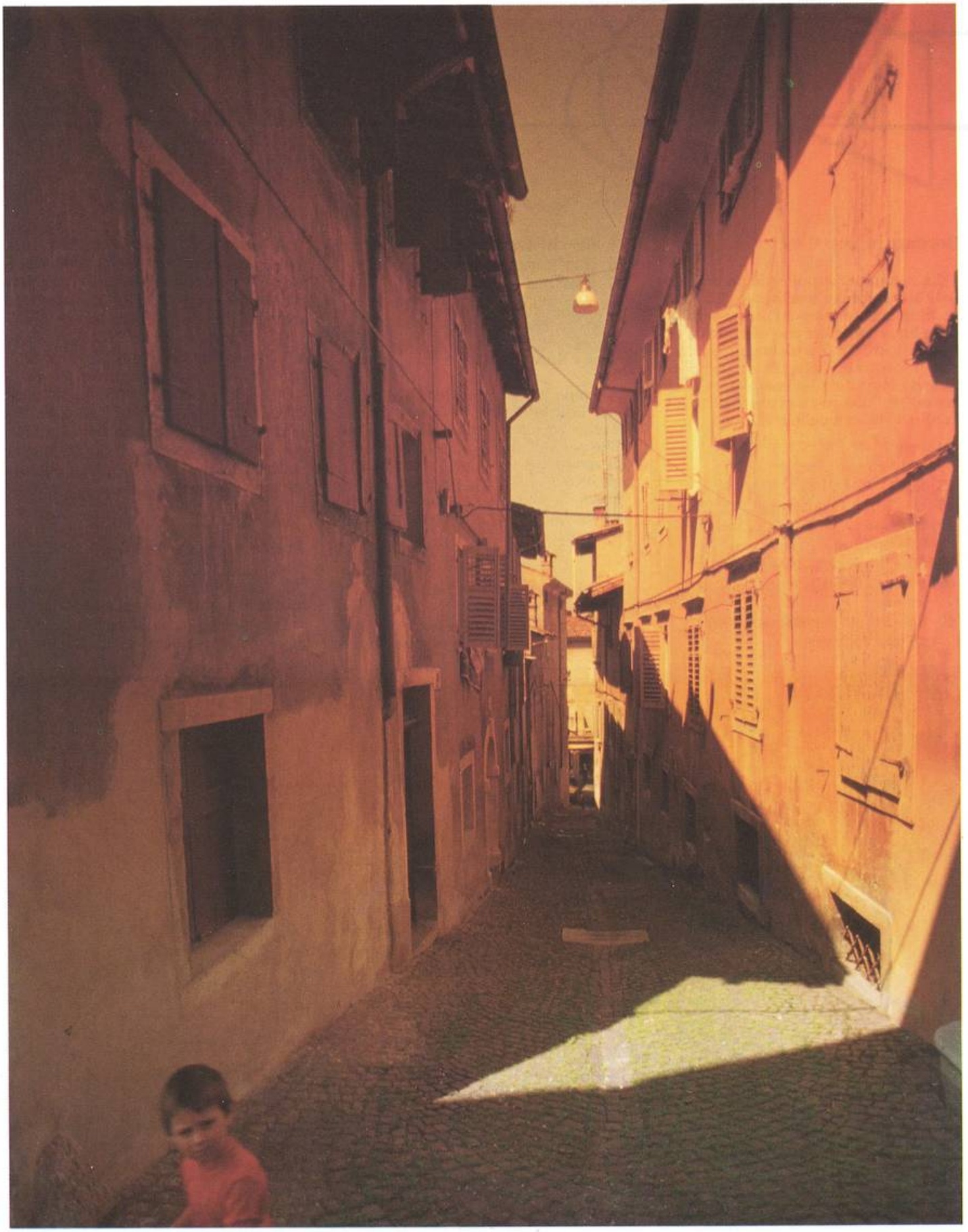
giornalieri;in queste immagini, ciò che più ci colpisce è l'attenta cura usata da Concina nell'utilizzo di vari filtri colore; tecnica che in pratica è il fulcro centrale di tutta la serie di diapositive pubblicate in questo numero della rivista.

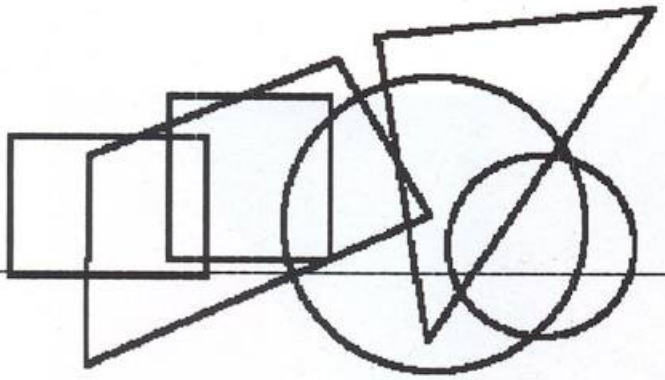
È con l'uso dei filtri che l'autore riesce a dare quelle calde tonalità nei colori delle sue immagini,quelle luci altrimenti impossibili da trovare, quel sapore quasi magico dei suoi paesaggi. Praticamente la nostra redazione è fatta di soli artisti; scherzi a parte speriàmo non vi dispiaccia questa nuova nostra iniziativa e oltretutto speriamo in una vostra partecipazione,molto gradita, nei prossimi numeri.

A chi interessa partecipare a "IMMAGINI", l'indirizzo è lo stesso:

Spett. redazione
"COMMODORE PROFESSIONAL"
IMMAGINI
Gruppo Editoriale Jackson
via Rosellini, 12
MILANO







Miscellanea

Bordatura per C-64

La seguente routine stampa una simpatica cornice sullo schermo. La routine inizia a stampare dal punto centrale della linea superiore e si estende da entrambi i lati fino a ricongiungersi sul punto centrale della linea inferiore. Quando la routine ha completato il suo lavoro, il cursore riappare nell'angolo in alto a sinistra dello schermo.

```
10 rem **********************
12 rem * cornice per c-64 *
15 rem **********************
20 c$="[ctrl 9]*":print"[shift clr]";
30 fori=1 to 19: printtab(20-i)c$tab
   (19+i)c$"[home]";: next
40 fori=1 to 24:printc$spc(38)c$;:next
50 printc$tab(38)c$"[crsr left] [shift inst]
   [crsr up]"
60 fori=1 to 19:printtab(i)c$tab(39-i)
   c$"[crsr up]": next
70 forl=217 to 242:pokel,peek(l)or128 :next
80 print"[home]": rem return
90 geta$: if a$=" "then 90
100 print"[shift clr]";: ifa$="r"then30
```

Geos e tasto restore

Quando passate dal Geos al Basic, il tasto Restore è stato ridefinito per effettuare un nuovo boot del Geos. È possibile però eliminare questa funzione in due modi. Nel primo modo basterà digitare POKE 792,71 e premere il tasto Return, la seconda opportunità è quella di spegnere il computer e riaccenderlo nuovamente.

Directory multiple per C-128

Questo è un ottimo sistema per posizionare fianco a fianco due directory su uno schermo in 80 colonne. In primo luogo, si effettui un Clear dello schermo, poi si carichi la directory del primo disco. Ora si riposizioni il cursore nella casella di Home (la prima in alto a sinistra) e si prema il tasto TAB cinque volte. Poi si premano i tasti ESC T per settare la parte superiore della nuova finestra. Si immetta nel drive il secondo disco e si richiami questa directory. Ora le directory appariranno fianco a fianco. Questa procedura è utile anche per comparare linee di programma durante la fase di programmazione.

Turbo cursore per C-64

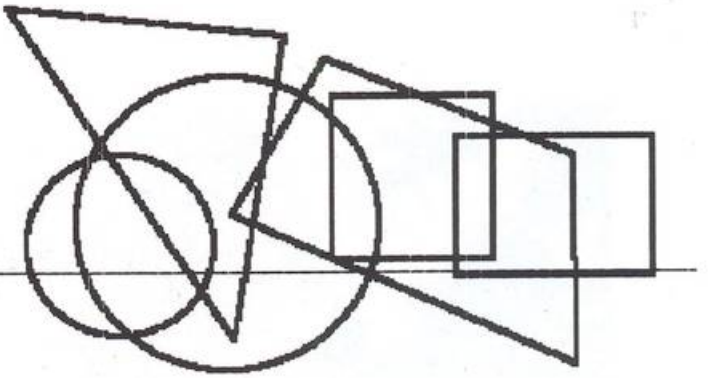
I seguenti comandi POKE faranno muovere molto velocemente il cursore sullo schermo, quest'opportunità è utile per programmi che richiedono un fre

```
cls schermo

   0 rem clr  schermo per
   5 rem c-128 40 colonne
   8 rem ----------------
  10 print"[2 home][clr]":char 1,15,1,
     "enter[spc]1-7"+chr$(13)
  15 getkey a$:z=instr("1234567",a$):
     on z goto 20,30,40,50,60,70,80:
     goto15
  20 gosub200:for a=0to19:window 19-a,
     0,20+a,24,1:next:goto10
  30 gosub200:for a=0to12:window 0,12-
     a,39,12+a,1:next:goto10
  40 gosub200:x=24:for a=35to0 step -5
     :x=x-3:window a,x,39,24,1:next:
     goto10
  50 gosub200:x=0:for a=4to39 step 5:x
     =x+3:window 0,0,a,x,1:next:goto10
  60 gosub200:for a=0to12:window 12-a,
     12-a,27+a,12+a,1:next:goto10
  70 gosub200:for a=0to39:window a,0,a
     ,24,1:next:window 0,0,39,24,0:
     goto 10
  80 gosub200:for a=0to39:window 39-a,
     0,39-a,24,1:next:window 0,0,39,24
     ,0:goto 10
 200 for x=1to25:print"**************
     ************************":next:
     return

mickey mouse

  10 rem **********************
  15 rem *   mickey mouse    *
  20 rem * c-128 40 colonne *
  30 rem **********************
  50 color0,2:color1,1:color2,8:color3
     ,3:color4,2:graphic3,1
  60 circle1,38,33,22,33:circle1,122,3
     3,22,33
  70 circle1,80,109,41,70,250,110:
     circle1,68,152,32,35,183,334
  80 circle1,92,152,32,35,26,177:
     circle1,80,70,18,28,105,255
  90 circle1,68,96,8,24:circle1,92,96,
     8,24
 100 circle1,68,105,4,12:circle1,92,10
     5,4,12
 110 circle1,66,80,9,20,270,25:circle1
     ,94,80,9,20,335,90
 120 circle1,58,92,10,34,213,340:
     circle1,102,92,10,34,20,147
 130 circle1,65,80,15,29,310,80:circle
     1,95,80,15,29,280,50
 140 circle1,80,109,30,55,118,242:
     circle1,60,152,12,20,290,358
```



quente utilizzo del cursore. Digitate semplicemente la seguente linea:

POKE 650,128: POKE 56325,10

Effettuando un poke della locazione 650 con il valore 128, viene abilitata la ripetizione dei tasti. È possibile mutare la velocità del cursore effettuando poke con valori diversi alla locazione 56325. Valori bassi daranno come risultato una maggiore velocità.

Numeri primi

Questo breve e rapido programma determina se un numero è un numero primo. L'analisi di numeri superiori al 10000 avrà bisogno di alcuni secondi; numeri di dimensioni maggiori avranno bisogno di circa un minuto di tempo. Il numero 9999973 richiede, per essere analizzato, di 40 secondi sul Commodore 64. Il programma funziona su tutti i modelli Commodore.

```
10 rem numeri primi
15 rem ************
20 print"[shift clr]"
30 input"[crsr giù]";x:z=
   sqr(x):y=1:ifx=0then end
40 if x/2=int(x/2)thenprint "[crsr
   giù]composto":goto30
50 fory=3tozstep2:d=x/y:i= int(d):s=d-i
60 if s>0then next
70 if s=0thenprint"[crsr giù]
   composto":goto30
80 print"[crsr giù] primo" :goto30
```

Cls schermo

Il programma che vi presentiamo contiene una serie di subroutine per ripulire, in modo simpatico, lo schermo del Commodore 128. La linea 200 del programma riempie lo schermo ed è stata inserita solamente per rendere visivamente l'effetto della routine. Non dovrete perciò inserirla nei vostri programmi.

Kentucky derby

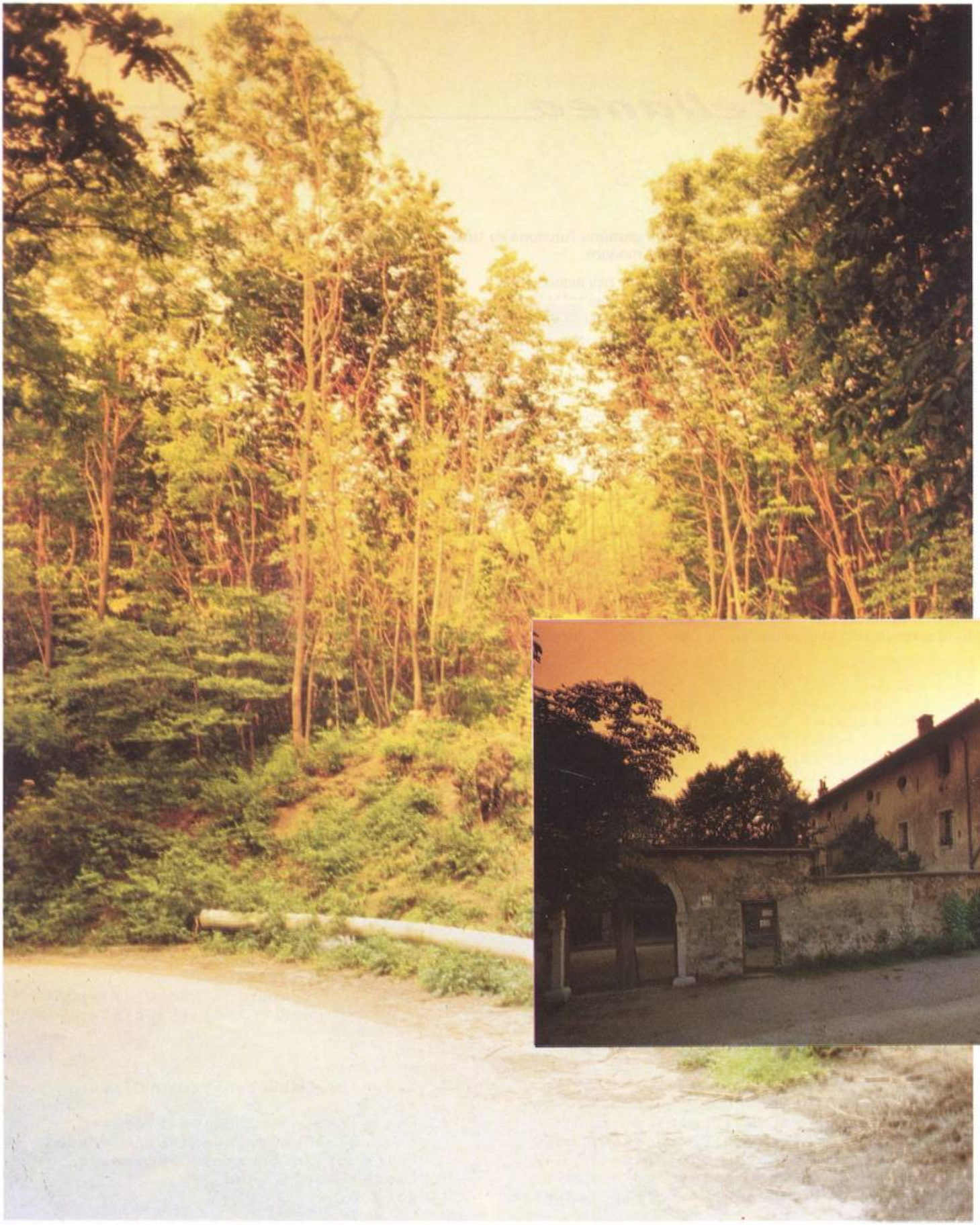
Non dovete recarvi in un ippodromo per lenire la vostra voglia di scommettere alle corse. Basterà il seguente programma che può essere eseguito sia sul C-64 che sul Commodore 128 in modo 40 colonne. Il programma visualizzerà una corsa con sette cavalli, generando le probabilità di vittoria per ciascun cavallo.

Il programma vi permette di fare delle puntate e vi mette a disposizione la somma di 500 dollari per le vostre scommesse; le gare si succederanno finché non avrete perso tutto il denaro o avrete arrestato il programma per mezzo del tasto Run-Stop.

```
150 circle1,100,152,12,18,2,70:circle
    1,80,160,35,38,332,28
160 circle1,80,160,13,22,90,270:
    circle1,76,178,8,9,310,70
170 circle1,86,178,8,9,325,35:circle1
    ,80,13,7,12,12
180 circle1,79,134,6,4:circle1,30,187
    ,30,20,90,200
190 circle1,130,187,30,20,140,270:
    circle1,32,200,37,40,75,90
200 circle1,128,200,37,40,270,285:
    paint1,77,199
210 paint1,42,109:paint1,80,147
220 paint1,80,165:paint3,65,198,1
230 paint3,95,198,1:paint3,80,175,1
240 paint1,40,33:paint1,120,33
250 paint1,68,105:paint1,92,105
260 paint2,70,152,1
```

```
kentucky derby

  0 rem kentucky derby per c-64 e c-1
    28 in modo 40 colonne
  1 m=500:poke53280,6:poke53281,6
  2 printchr$(142)"[clr][4 spc]
    [ctrl<8>][rvon][2 spc]kentucky
    [2 spc]derby[2 spc][rvoff]"
  3 foro=1to7:o(o)=int(rnd(1)*26+1):
    print"[cmdr<5>][rvon][spc]cavallo
    "o"[right][spc][rvoff][spc]valuta
    to"o(o)"a[spc]1"
  4 print"[ctrl<2>][spc]
    [shift<uparrow>][down]":next:foro
    =1to7:p(o)=0:next:print:print"
    [2 up][rvon]tu[spc]hai[spc]$"m
  5 input"[2 spc]-su[spc]quale[spc]ca
    vallo[spc]punti";h:ifh>7orh<1then
    print"[2 up]":goto5
  6 input"[2 spc]-quanto[spc]denaro
    [spc]punti";b:ifb>mthenprint"
    [2 up]":goto6
  7 x=int(rnd(1)*7+1):p(x)=p(x)+1
  8 print"[home]";:ford=1tox*3-1:
    print"[down]";:next:forp=1top(x):
    print"[spc]";:next:print"
    [ctrl<2>][shift<uparrow>]"
  9 ifp(x)=39thenprint"[home]"tab(24)
    "[rvon]vince[2 spc]il[spc]#"x:
    goto11
 10 goto 7
 11 ifx<>hthen13
 12 m=m-b+b*o(x):forq=1to3000:next:
    goto2
 13 m=m-b:forq=1to3000:next:ifm<=0
    thenprint"[clr][rvon]*[spc]*[spc]
    hai[spc]finito[spc]i[spc]soldi
    [spc]*[spc]*":end
 14 goto 2
```



C64 & programmazione strutturata

La potenza del Pascal con la facilità del BASIC

Nel numero scorso avevamo già avuto qualche approccio con questa tecnica universale. In particolare vi sono stati offerti alcuni suggerimenti per implementare una programmazione di tipo strutturato sul vostro C64, senza ricorrere a particolari routine di supporto; vi ricordiamo che in quella occasione era nostra intenzione farvi perdere il 'vizio' nell'abusare dei GOTO, appunto per introdurvi, in questa seconda parte, nell'ambiente rigoroso delle strutture. Nel discorso che seguirà, verranno introdotte alcune nuove istruzioni, acquisite da un linguaggio pseudo-Pascal, utilizzabili dall'utente di C64; queste nuove potenzialità permettono di scrivere programmi strutturati, utilizzando direttamente le strutture, senza cioè ricorrere a trucchi d'implementazione. L'espansione viene realizzata ovviamente con un utility in linguaggio macchina, di cui viene riportato il listato nel presente articolo.

L'argomento 'programmazione strutturata' è centro di dibattito in tutti gli ambienti, inseriti nel mondo informatico.

Non a caso, nell'ambito scolastico, non c'è sezione informatica che non metta l'accento sull'importanza di questa tecnica di programmazione.

Ma quest'ultima non si limita comunque al solo campo universitario; gran parte delle grosse aziende richiedono ai loro dipendenti, addetti ai sistemi di informazione, la completa conoscenza e utilizzo di tecniche strutturate di programmazione, nell'adempiere il loro lavoro.

Bene, con tutte queste enfasi nei riguardi della programmazione strutturata, non è insensata l'idea di escogitare un possibile metodo per permettere, anche ai numerosi utenti di C64, di impratichirsi con tutta comodità di queste preziose tecniche, senza ricorrere a sofisticati e costosi pacchetti software di espansione.

Sfortunatamente è assai complicato, ma non proprio impossibile, implementare programmi strutturati su C64, a causa delle limitazioni del BASIC 2.0.

La routine che vi presentiamo, dà la possibilità di usufruire delle strutture più comunemente adottate nei linguaggi di questo tipo. Se saprete sfruttare a pieno queste nuove potenzialità, seguendo inoltre le modalità della programmazione strutturata, state certi che i vostri programmi risulteranno più eleganti, flessibili, facili da leggere e da modificare.

Prima di descrivere una per una le nuove istruzioni, forse è meglio procedere a passi successivi e discutere più dettagliatamente su cosa si intende dire, quando si parla di programmazione strutturata.

I due più importanti concetti riguardanti questa tecnica sono la 'progettazione TOP-DOWN'e la 'programmazione MODULARE'.

TOP-DOWN

Molto concisamente, la progettazione di tipo top-down si riferisce all'idea di considerare un certo lavoro (anche complesso) e quindi di scomporlo in piu' semplici sotto-lavori. Quest'ultimi possono essere a loro volta considerati come lavori, perciò semplificabili anch'essi; completata questa operazione di successive scomposizioni, otteniamo così l'intero lavoro suddiviso in semplici sotto-lavori elementari.

L'aver usato il termine generico 'lavoro' fa presagire che la tecnica top-down non dipende strettamente dall'ambiente della programmazione su computer, ma può essere estesa su un campo ben più vasto.

La progettazione top-down può risultare utile per l'implementazione di programmi di qualsiasi linguaggio, incluso l'assembler.

L'idea essenziale è quella di poter scrivere qualsiasi programma, incominciando dal definirne prima uno, assai generico, composto da una serie di richieste specifiche, scritte senza fare molto caso al linguaggio, ma seguendo a grandi linee il flusso che il programma ultimato dovrà assumere. Questa specie di prototipo viene di solito chiamato programma principale (o MAIN) e consiste in definitiva in una serie di dichiarazioni iniziali e quindi di chiamate a subroutine inserite all'interno di strutture, quali la sequenziazione, il blocco decisionale ed il ciclo.

Ecco di seguito riportato un semplice esempio di progettazione (o programmazione) top-down. Il problema, visto nella sua totalità richiede l'inserimento di caratteri all'interno di un vettore; quindi calcola le occorrenze di lettere, cifre e caratteri speciali.

Da una prima analisi del problema, lo possiamo scomporre nei seguenti quattro sotto-problemi che, in fase di implementazione su computer, risulteranno essere le chiamate alle subroutine da parte del programma principale. Si dovrebbe discutere anche del trasferimento dei parametri nella fase di chiamata, ma questo ci porterebbe troppo fuori da quelli che sono i nostri obiettivi e introderrebbe inutili complicazioni.

Da una prima semplificazione, il problema potrebbe apparire nel seguente modo:

```
Definizione variabili
REPEAT
Inserimento dato nel vettore
UNTIL non ci sono più dati
Calcolo occorrenze mediante ciclo For
Stampa occorrenze
```

A questo punto ciascuna subroutine può essere a sua volta meglio specificata, ottenendo:

```
Definizione variabili.
----------------------
Definizione:
    variabile DATO di tipo carattere
    vettore VETT di caratteri
    variabili
    CONTOT,K,CONTLETT,CONTCIFR,
            CONTSPEC di tipo intero
Inizializzazione:
  CONTOT=CONTLETT=CONTCIFR
  =CONTSPEC=0

Inserimento dato nel vettore.
-----------------------------
Leggo dato
Incrementa CONTOT
VETT[CONTOT]<-- DATO

Calcolo occorrenze mediante ciclo For.
-----------------------------
FOR K=1 TO CONTOT
   DATO<--VETT[K]
   IF (DATO<'z'and DATO>'à)
      THEN Incrementa CONTLETT
      ELSE IF (DATO<'9' and
            DATO>'0')
            THEN      Incrementa
            CONTCIFR
            ELSE      Incrementa
            CONTSPEC
          (ENDIF)
  (ENDIF)
NEXT K (ENDFOR)

Stampa occorrenze.
------------------
Stampa CONTOT,CONTLETT,CONTCI-
FR,CONTSPEC
```

La parola 'modulo' si riferisce proprio ad una subroutine che fa parte dell'intero programma; quindi la programmazione modulare, come vedremo, è proprio quella che fa uso di MAIN e subroutine, ciascuna indipendente dalle altre.

Programmazione modulare

Lo scopo principale di questa tecnica è quello di produrre un programma, il quale risulti composto da ben definiti moduli; con ciò vogliamo dire che se aggiungiamo o togliamo uno di questi moduli, tali operazioni non influiscono minimamente sulle altre parti del programma.

Ciascun modulo ha come caratteristica quella di possedere un unico punto di entrata ed un unico punto di uscita, il che rende conto sull'indipendenza dagli altri.

Da quanto detto, in un programma scritto a moduli, qualsiasi operazione effettuata su uno di essi (creazione, modifica, cancellazione) non richiede alcun controllo sul resto del programma.

La progettazione modulare risulta molto importante per quanto riguarda i programmi di tipo manageriali e di marketing, i quali necessitano di continue modifiche e revisioni, a seconda di come varia il mercato.

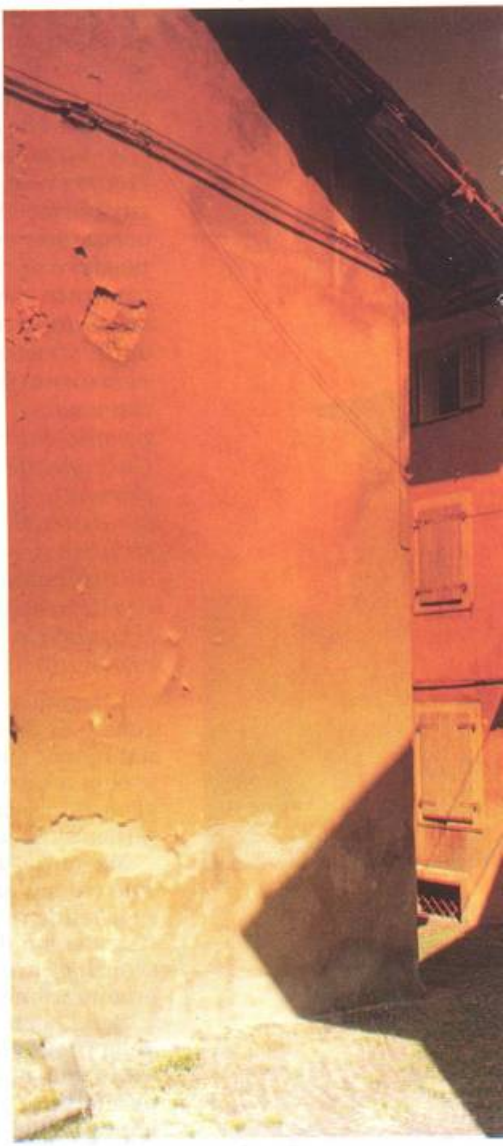
Un utente, dal canto suo, può sfruttare la programmazione modulare per rendere i propri programmi il più possibile compatibili con altre macchine, raggruppando tutte le istruzioni dipendenti dal proprio computer all'interno di moduli distinti, piuttosto che averle sparpagliate da un capo all'altro del programma. Sarà quindi più semplice implementare ciascun modulo su un'altra macchina, che tradurre in un unico blocco, l'intero programma.

Ad esempio, supponiamo di aver scritto un programma su C64, di notevoli dimensioni, che calcola la tassa sul reddito; supponiamo inoltre che successivamente vogliamo implementare questo programma su un PC IBM. Se è stato ben progettato, allora probabilmente si dovrà ridefinire soltanto il modulo di inizializzazione (che effettua l'apertura dei file, definisce i parametri dipendenti dalla macchina, ecc.), il modulo di lettura (dedicato alla lettura da disco) ed il modulo di scrittura (che visualizza i dati su video e stampante). Tutti gli altri moduli (dedicati principalmente ai calcoli) non necessitano invece di particolari modifiche. Riferendoci al programma dell'esempio, se si verificano cambiamenti sulle leggi riguardanti le tasse, sarà sufficiente dare una controllata ai soli moduli che compiono calcoli dipendenti dalle leggi in vigore.

Programmazione strutturata

I due concetti appena analizzati convergono in un unico obiettivo: la programmazione strutturata.

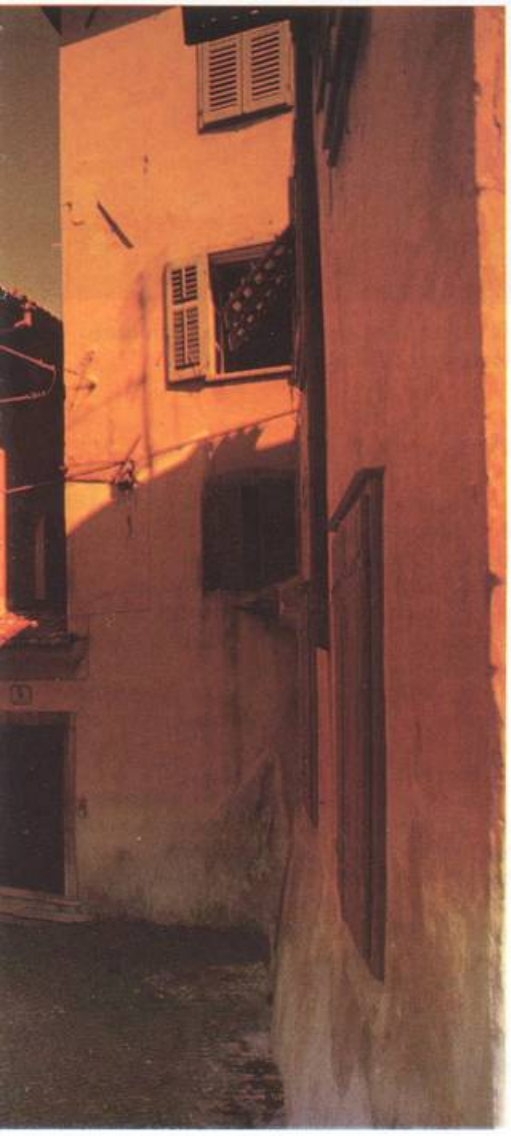
La programmazione top-down e modulare sono comunque un aspetto puramente concettuale della programmazione strutturata: le prime due riguardano la fase di progetto di un programma, la terza si riferisce all'implementa-



zione del progetto su elaboratore, secondo alcune regole per migliorare la leggibilità e la possibilità di correzioni. Queste regole richiedono severe limitazioni all'uso dell'istruzione GOTO, offrono la possibilità di indentazione del listing (per isolare blocchi appartenenti a cicli o altre strutture dal resto del programma) ed infine, come regola di carattere pratico, il consiglio di adottare nomi significativi per i moduli richiamabili dal MAIN.

Chiaramente il BASIC 2.0 del C64 non è adatto al tipo di programmazione appena descritta; ed è questa, naturalmente, la ragione per cui vi presentiamo un utility in LM, che espande il set di istruzioni della ROM BASIC, aggiungendovi l'opportunità della strutturazione.

Ovviamente non avrete fra le mani il Pascal, ma nonostante ciò riuscirete a lavorare seguendo le principali regole della programmazione strutturata. I nuovi comandi, che verranno introdotti tra poco, sono abbastanza simili a quelli incontrati in linguaggi, quali Pascal (soprattutto), COBOL e C.



Per quanto riguarda le strutture, sono state introdotte quelle che meglio si adattano ai concetti della progettazione strutturata. Esse sono:

WHILE-WEND
IF-THEN-ELSE
REPEAT-UNTIL
CALL-PROC

Sebbene sia fuori luogo presentare un intero corso su come e quando utilizzare le strutture, si è pensato di dare una breve descrizione di ciascuna di esse e completare il tutto con qualche esempio.

Siamo certi che chi è interessato ad adottare le strutture nella sua normale programmazione, non dovrà aspettare molto tempo per poter manipolarle con naturalezza e, con l'esperienza, il loro valore verrà sempre di più apprezzato.

Chi di voi ha già fatto esperienza con linguaggi come il Pascal su C64, apprezzerà ancora di più la comodità di avere a disposizione le strutture direttamente dall'ambiente interpretativo del BASIC. La facilità di effettuare correzioni e modifiche da BASIC, taglia drasticamente i tempi che, possedendo un pacchetto software di espansione, sarebbero necessari per caricare l'editor, il compilatore, il linker e gli eventuali moduli, in versione oggetto. Dal BASIC queste operazioni non sono necessarie, anche se i pregi della compilazione si fanno sentire in fase di esecuzione del programma.

Usare i nuovi comandi

Per accedere alle nuove possibilità offerte, dovete scrivere in memoria il programma caricatore in BASIC (listato 1) e quindi attivare l'espansione con il comando: SYS 49152

Ecco di seguito riportata la lista dei nuovi comandi e strutture, corredati di descrizione e relativo esempio.

EDIT

Questo comando permette di inserirsi nell'editor del C64, con la novità che in questo caso il carattere 'spazio' viene considerato significativo; questo acconsente l'indentazione durante l'inserimento dei programmi (nel BASIC questo non è normalmente possibile).

L'indentazione, che fa parte della sintassi della programmazione strutturata, si rivela particolarmente utile per isolare blocchi di istruzioni che stanno sullo stesso livello, come quelle inserite all'interno di un ciclo. Ancora più preziosa si dimostra quando ci sono più cicli nidificati; in questo caso la leggibilità è d'obbligo.

ATTENZIONE! Quando si è in ambiente editor, è vietato l'uso degli spazi quando si eseguono comandi in modo diretto, a meno che non siano fra apici. Una tale dimenticanza causa un errore di sintassi. Inoltre, non si devono collocare spazi dopo il numero di linea, quando si inseriscono le linee di programma.

L'uscita dall'ambiente editor avviene automaticamente con il comando RUN.

KILL

Questo comando serve per uscire dall'editor, senza però dover mandare in esecuzione il programma.

BASIC2

Questo comando disabilita la funzione dell'espansione e ritorna il controllo al BASIC 2.0. Se si desidera riattivarla, basta rieseguire il comando SYS 49152.

CALL-PROC

L'istruzione CALL lavora esattamente come una GOSUB, con la differenza che in questo caso si può chiamare una routine (modulo) per nome, anziché per numero di linea. Questa opportunità permette di assegnare a ciascun modulo un nome che sia significativo per quanto riguarda la sua funzione.

La parola PROC viene utilizzata per definire l'inizio di un modulo (PROCedure) e quindi questa deve essere la prima istruzione. Attenzione a non inserire spazi prima della PROC (vedi errore #6).

Una singola procedura termina sempre con la dichiarazione RETURN.

```
10  FOR I=1 TO 10
20  CALL OUTPUT
30  NEXT I
40  END
50  :
60  PROC OUTPUT
70  PRINT"OUTPUT CALL # ";I
80  RETURN
```

REPEAT-UNTIL

Questa struttura è abbastanza simile al ciclo FOR-NEXT con l'unica eccezione che invece di ripetere il blocco di istruzioni per un certo numero di volte prefissato, in questo caso la ripetizione avviene fino a che una condizione specificata diventa vera.

L'esempio simula il ciclo FOR, visualizzando i numeri dall'uno al dieci.

```
10 REPEAT
15 X=X+1
20 PRINT X
30 UNTIL X>=10
```

WHILE-WEND

Un uso intelligente di questa struttura vi permette di togliere un mucchio di GOTO dai vostri programmi. Tale ciclo ripete il blocco di istruzioni al suo interno, mentre una condizione specificata permane vera; cioè si esce dal ciclo non appena la condizione diventa falsa.

Come si era già detto lo scorso numero, l'unica vera differenza fra questo ed il ciclo precedente sta nella posizione del test condizione: per il ciclo REPEAT-UNTIL il test è posizionato alla fine del blocco di istruzioni (come per il FOR), mentre per il ciclo WHILE-WEND il test è all'inizio del blocco. Ciò significa che con il REPEAT-UNTIL si passa almeno una volta per il corpo del blocco, quindi, come si vede dall'esempio, non è adatto a 'simulare' un GOTO.

```
10 OPEN 2,8,2,"TEST.FIL,S"
20 WHILE ST=0
30 GET#2,A$
40 PRINT A$;
50 WEND
```

Tale routine legge un file sequenziale e ne visualizza il contenuto.

EXIT

Certamente saprete che, con il BASIC 2.0 come per altri, non è consigliabile uscire con un GOTO da un ciclo FOR o da una subroutine. Per la stessa ragione ne segue che è meglio evitare l'uso del GOTO per uscire dai cicli WHILE-WEND e REPEAT-UNTIL, e anche da PROC.

Se tuttavia si verifica una condizione straordinaria, per cui si deve uscire inderogabilmente da una delle strutture, si deve usare il comando EXIT al posto del GOTO. EXIT lavora esattamente come il GOTO, con la differenza che, prima di effettuare il salto, mette a posto lo stack, per evitare spiacevoli inconvenienti durante il seguito dell'esecuzione.

Si può definirlo un comando intelligente, nel senso che non occorre comunicargli da quale tipo di struttura si intende uscire; è suo compito capire se si tratta di FOR-NEXT, REPEAT-UNTIL, WHILE-WEND, GOSUB o PROC.

Si può uscire da una sola struttura alla volta, cioè se ci troviamo all'interno di tre cicli nidificati, allora eseguendo EXIT ci troveremo all'uscita del terzo ciclo, cioè nel corpo del secondo. Tale caratteristica comporta una maggiore flessibilità del comando e implica una sola cancellazione alla volta nello stack.

```
10 X=1: INPUT"INSERISCI UN INTERO";K
20 FOR I=K TO 1 STEP -1
30 X=X*K*RND(0)
40 IF ABS(X)>1E7 THEN EXIT 90
50 PRINT"X=";X
60 NEXT I
70 PRINT"RISULTATO FINALE:";X
80 END
90 PRINT"****ERRORE****"
95 PRINT"NUMERO TROPPO ELEVATO"
99 PRINT"PROVA DI NUOVO": GOTO 10
```

IF-THEN-ELSE

È una struttura che sta su una sola linea; rispetto al test del BASIC 2.0, questa ha anche la clausola ELSE, che permette di eseguire delle istruzioni anche quando la condizione è falsa. IF ABS(X)<1 THEN CALL NEWTON: ELSE CALL GAUSS

Il significato di questa struttura è: 'se il valore assoluto di X e' minore di uno, allora viene chiamato il modulo NEWTON; altrimenti viene chiamato il modulo GAUSS.'

Messaggi di errore

Siccome questa estensione del BASIC è stata intesa per essere utilizzata nella programmazione di tipo 'real-lifè, allora all'introduzione di nuovi comandi deve corrispondere una adeguata diagnostica, circa i nuovi errori che possono capitare.

Ecco una lista degli otto messaggi di errore, e delle probabili cause relative a ciascuno di essi.

#1 UNTIL WITHOUT REPEAT

Si verifica sia quando si omette l'istruzione REPEAT, sia quando si sono nidificate, in modo improprio, le strutture. Può anche essere causato da un irregolare uso dell'istruzione EXIT.

#2 WEND WITHOUT WHILE

Analogo all'#1.

#3 WHILE WITHOUT WEND

Si verifica quando ci sono meno WEND che WHILE nel programma. Vedi anche #1.

#4 MISSING LOGICAL EXPRESSION

La parola WHILE o anche UNTIL non

è seguita da un'espressione logica.

#5 NO STRUCTURE TO EXIT

Si è tentato l'EXIT al di fuori di una struttura. Può talvolta indicare problemi nel flusso del programma; ad esempio quando c'è un GOTO la cui label è all'interno di una struttura.

#6 PROCEDURE NOT FOUND

Si verifica sia se non esiste una procedura col nome specificato, sia se la parola PROC non è la prima della linea. Controllate inoltre che non vi siano spazi prima di PROC.

#7 OUT OF MEMORY

Si verifica se ci sono troppe strutture nidificate o anche se ci sono problemi nel flusso del programma. Di solito può accadere quando si esce in modo improprio dalle strutture.

#8 PROC WITHOUT CALL

Probabilmente si è dimenticato l'END a fine MAIN, cioè prima della serie dei moduli. Indica pure problemi di flusso in generale.

Programma di esempio

Il programma demo, che riportiamo assieme al caricatore BASIC, non compie nulla di estremamente originale e non è certo il miglior esempio di programmazione strutturata; comunque fa uso di tutti i nuovi comandi di espansione introdotti.

Quando lo si manda in esecuzione, sarà chiesto di scegliere fra le due opzioni, la lettura di un file su disco e la creazione di un file su disco. Se si sceglie la lettura, viene richiesto il filename e quindi tale file viene visualizzato. Se si sceglie la creazione, viene prima richiesto il filename e poi le linee di caratteri, da inserire nel file sequenziale. Non ci sono limitazioni sui tipi dati che si possono inserire; ogni carattere che appare sulla tastiera può essere ritenuto valido.

Quando pensate di aver terminato la fase di inserimento, allora battete semplicemente due volte il tasto Return per chiudere il file.

In conclusione

Prima di concludere, vi diamo alcuni spunti per rendere ancora più potente e completa l'espansione.

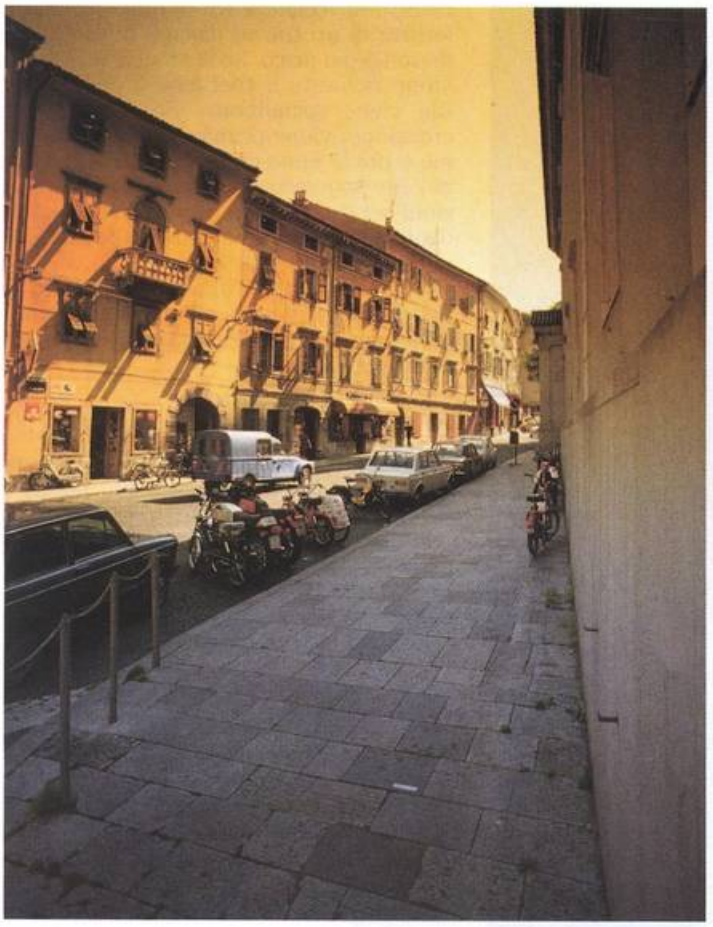
Il primo riguarda, come avrete già immaginato, il potenziamento della struttura IF-THEN-ELSE che, per evitare complesse ripetizioni del codice, abbiamo relegato ad una sola linea. Per far ciò si necessita anche l'introduzione della dichiarazione ENDIF, la quale determina la fine del blocco IF. Un esempio:

```
10  IF X>10 THEN
20  PRINT"X È MAGGIORE DI 10"
30  PRINT"BUON PUNTEGGIO!"
40   ELSE
50  PRINT"X È MINORE O UGUALE A 10"
60  PRINT"PECCATO!"
70  ENDIF
```

Se avete intenzione di implementare tale struttura per conto vostro, eccovi una traccia su cosa deve fare la routine in corrispondenza di ciacuna chiave IF, ELSE, ENDIF. IF- Analizza la condizione. Se vera procedere normalmente. Se falsa cercare la chiave ELSE e continuare l'esecuzione dopo quella. ELSE- Se viene eseguito un ELSE, significa che si è giunti alla fine del blocco IF; quindi bisogna cercare la dichiarazione ENDIF e continuare l'esecuzione dopo di essa. ENDIF- ENDIF non fa altro che segnare la fine del blocco IF. È sufficiente eseguire un RTS.

Se c'è qualcuno veramente ambizioso, allora una modifica alla routine relativa a CALL-PROC che includa il trasferimento di parametri, potrebbe risultare un notevole potenziamento alla routine da noi presentata.

```
100 rem ** caricatore per estensione
101 rem ** basic strutturato
102 rem *** sys 49152 per attivare
103 :
104 for adr=49152 to 50274: read ml
105 cs=cs+ml: poke adr,ml: next
106 if cs<>136922 then print"errore nei data"
107 :
108 data 169, 49,141,  8,  3,169,192,141
109 data   9,  3,169, 17,160,192, 76, 30
110 data 171, 62, 32, 83, 84, 82, 85, 67
111 data  84, 85, 82, 69, 68, 32, 67, 79
112 data  77, 77, 65, 78, 68, 83, 32, 69
113 data  78, 65, 66, 76, 69, 68, 46, 13
114 data   0, 32,115,  0, 32, 67,192, 76
115 data 174,167, 76, 59,169, 32,  5,193
116 data  76,180,192,201, 39,240,243,201
117 data 139,208,  3, 76, 44,193,201,138
118 data 240,235,170, 48, 95,160, 10,140
119 data 186,192,201, 87,208,  8,160,  1
120 data 177,122,201,128,240, 52,169,  0
```

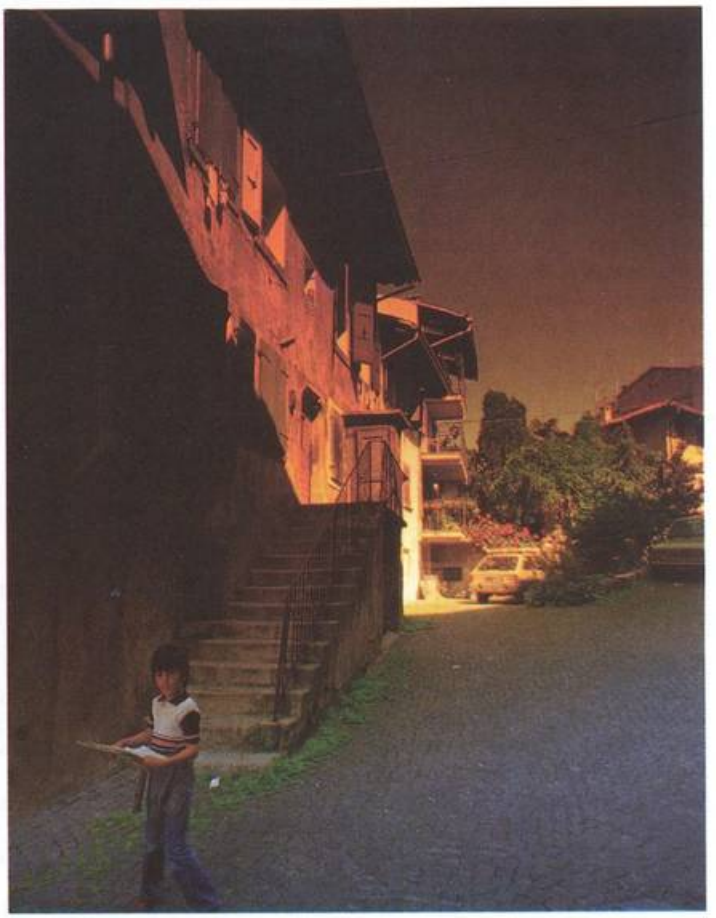


```
121 data 141,186,192,170,168,136,200,189
122 data 187,192,240, 64,232,209,122,208
123 data   2,240,243,202,189,187,192, 16
124 data   8, 41,127,209,122,240, 19,208
125 data   8,232,189,187,192,240, 37, 16
126 data 248,232,238,186,192,160,255, 76
127 data 110,192,152, 24,101,122,133,122
128 data 144,  2,230,123,173,186,192, 10
129 data 170,189,235,192, 72,189,234,192
130 data  72, 76,115,  0, 32,121,  0, 76
131 data 237,167,  0, 82, 69, 80, 69, 65
132 data 212, 85, 78, 84, 73,204, 87, 72
133 data  73, 76,197, 69, 88, 73,212, 67
134 data  65, 76,204, 80, 82, 79,195, 69
135 data  76, 83,197, 69, 68, 73,212, 75
136 data  73, 76,204, 66, 65, 83, 73, 67
137 data 178,  0,129,193,155,193,217,193
138 data 188,194,237,194,136,195,126,193
139 data 255,192,  4,193,  9,193, 11,194
140 data 169,255,133,129, 96,169, 32,133
141 data 129, 96,169,228,141,  8,  3,169
142 data 167,141,  9,  3,169, 28,160,193
143 data  76, 30,171, 96, 62, 32, 67, 77
144 data  68, 83, 32, 68, 73, 83, 65, 66
145 data  76, 69, 68,  0, 32,115,  0, 32
146 data 158,173, 32,121,  0,201,137,240
147 data   6,169,167, 32,255,174,165, 97
148 data 208,  7, 32, 99,193,170,208, 22
149 data  96, 32,121,  0,176,  3, 76,160
150 data 168,165,122, 56,233,  1,133,122
151 data 176,  2,198,123,160,  0,104,104
152 data 108,  8,  3, 32,  6,169, 72, 32
153 data 251,168,104,240, 13,162,  3, 32
154 data 115,  0,221,123,193,208,236,202
155 data  16,245, 96, 69, 83, 76, 69, 76
156 data  59,169,169,  3, 32,251,163, 32
157 data 248,168,165,123, 72,165,122, 72
158 data 165, 58, 72,165, 57, 72,169,231
159 data  72, 76,174,167, 32,138,163,154
160 data 201,231,208, 40, 32,121,  0,240
161 data  44, 32,158,173,186,138, 24,105
162 data   5,170,168,165, 97,208, 19,162
163 data   1,136,185,  1,  1,149,122,185
164 data 255,  0,149, 57,202, 16,242, 76
165 data 174,167,154, 96,169,  0, 44,169
166 data   1, 44,169,  2, 44,169,  3, 76
167 data 189,195, 32,121,  0,240,246,169
168 data   3, 32,251,163,165,122,141,185
169 data 194,165,123,141,186,194, 32,158
170 data 173,165, 97,208,  3, 76,101,194
171 data 173,186,194, 72,173,185,194, 72
172 data 165, 58, 72,165, 57, 72,169,235
173 data  72, 76,174,167, 32,138,163,154
174 data 201,235,208,187, 32,121,  0,208
175 data 185,165,123,141,186,194,165,122
176 data 141,185,194,165, 58,141,188,194
177 data 165, 57,141,187,194,186,138, 24
178 data 105,  5,170,142,183,194,168,162
179 data   1,136,185,  1,  1,149,122,185
180 data 255,  0,149, 57,202, 16,242, 32
181 data 158,173,165, 97,240,  3, 76,174
182 data 167,174,183,194,154,162,  1,189
183 data 185,194,149,122,189,187,194,149
184 data  57,202, 16,243, 96,169,  0, 72
185 data  32,248,168, 32,121,  0,170,240
186 data  18, 32,115,  0,170,240, 12, 32
187 data 142,195,240, 34, 32,145,195,240
188 data  35,208,229,160,  2,177,122,208
189 data   3, 76,210,193,200,177,122,141
190 data 187,194,200,177,122,141,188,194
191 data  32,251,168, 76,113,194,104,240
192 data   9, 76,104,194,169,235, 72, 76
193 data 104,194,173,187,194,133, 57,173
194 data 188,194,133, 58, 76,248,168,  0
195 data   0,  0,  0,  0,  0,104,104,104
196 data 201,129,240, 17,201,141,240, 16
197 data 201,231,240, 12,201,235,240,  8
198 data 169,  4, 76,189,195,169, 19, 44
199 data 169,  6,141,184,194,186,138, 24
200 data 109,184,194,170,154, 32,121,  0
201 data  32,160,168, 76,174,167,169,  3
202 data  32,251,163,165,123, 72,165,122
203 data  72,165, 58, 72,165, 57, 72,169
204 data 141, 72, 32, 24,195,162,  1,181
205 data 251,149,122,181, 97,149, 57,202
206 data  16,245, 32,248,168, 76,174,167
207 data 165, 43,133,253,165, 44,133,254
208 data 165,253,133,251,165,254,133,252
209 data 160,  1,177,251,208,  5,169,  5
210 data  76,189,195,133,254,136,177,251
211 data 133,253,160,  4,177,251, 32,159
212 data 195,208,221,160,  3,177,251,133
213 data  98,136,177,251,133, 97,160,  7
214 data 200,177,251,201, 32,240,249,152
215 data  24,101,251,133,251,144,  2,230
216 data 252,160,255,200,177,122,240, 24
217 data 201, 58,240, 20,201, 32,208,  9
218 data 230,122,208,  2,230,123, 76,100
219 data 195,209,251,240,230, 76, 32,195
220 data 177,251,240,  4,201, 58,208,152
221 data  96,169,  6, 76,189,195,162,  4
222 data  44,162,  7,164,122,132,251,164
223 data 123,132,252,160,255,208,  4,162
224 data 255,160,  3,200,232,189,175,195
```

```
225 data 240,  4,209,251,240,245, 96, 80
226 data  82, 79, 67,  0, 87,128,  0, 87
227 data  72, 73, 76, 69,  0, 10,170,189
228 data 202,195,133, 34,189,203,195, 76
229 data  69,164,216,195,236,195,254,195
230 data  16,196, 42,196, 62,196, 81,196
231 data  85, 78, 84, 73, 76, 32, 87, 73
232 data  84, 72, 79, 85, 84, 32, 82, 69
233 data  80, 69, 65,212, 87, 69, 78, 68
234 data  32, 87, 73, 84, 72, 79, 85, 84
235 data  32, 87, 72, 73, 76,197, 87, 72
236 data  73, 76, 69, 32, 87, 73, 84, 72
237 data  79, 85, 84, 32, 87, 69, 78,196
238 data  77, 73, 83, 83, 73, 78, 71, 32
239 data  76, 79, 71, 73, 67, 65, 76, 32
240 data  69, 88, 80, 82, 69, 83, 83, 73
241 data  79,206, 78, 79, 32, 83, 84, 82
242 data  85, 67, 84, 85, 82, 69, 32, 84
243 data  79, 32, 69, 88, 73,212, 80, 82
244 data  79, 67, 69, 68, 85, 82, 69, 32
245 data  78, 79, 84, 32, 70, 79, 85, 78
246 data 196, 80, 82, 79, 67, 32, 87, 73
247 data  84, 72, 79, 85, 84, 32, 67, 65
```

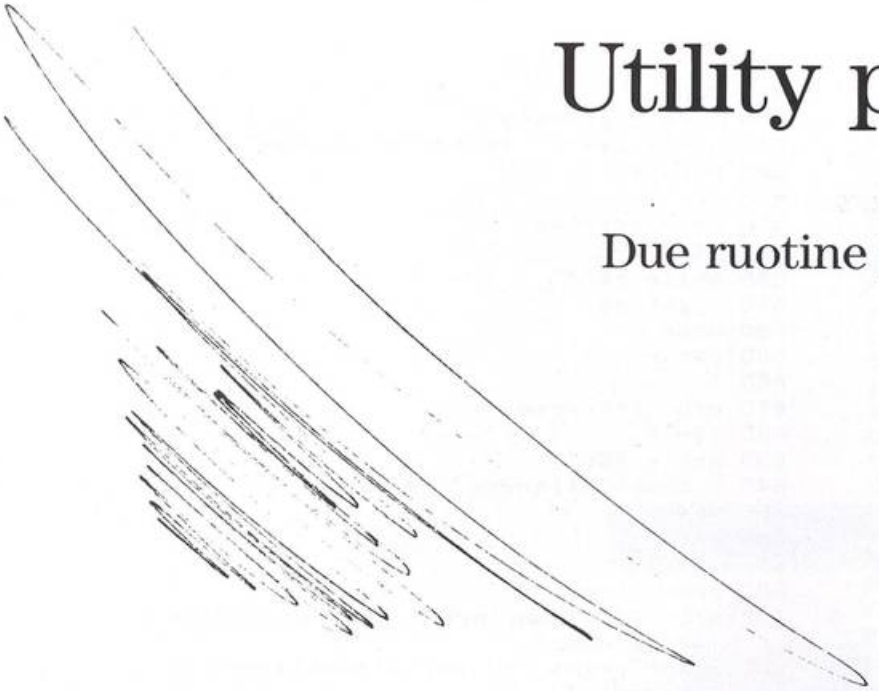
Programma esempio

```
100 '
110 'demo strutturato
120 '
130 call prompt
140 while a$<>"x"
150   if a$="c" then call create:
                 else call read
160   call prompt
170 wend
180 end
190 :
200 proc create
210 call get-name
220 open 2,8,2,f$+",w"
230 print"inserisci i dati..."
240 call get-key
250 repeat
260   print a$;
270   print#2,a$;
280   repeat
290     call get-key
300     print a$;
310     print#2,a$;
320   until a$=chr$(13)
330   call get-key
340 until a$=chr$(13)
350 print#2: close2
360 return
370 :
380 proc read
390 call get-name
400 open 2,8,2,f$
410 while st=0
420   get#2,a$
430   print a$;
440 wend
450 close 2
460 return
470 :
480 proc get-valid-key
490 repeat
500   call get-key
510 until a$="c" or a$="d" or a$="x"
520 return
530 :
540 proc get-key
550 a$=""
560 while a$=""
570   get a$
580 wend
590 return
600 :
610 proc get-name
620 f$=""
630 while f$=""
640   input"filename";f$
650 wend
660 return
670 :
680 proc prompt
690 print"[2 Down]premi 'c' per creare i
    l file dati"
700 print"premi 'd' per visualizzare il
    file dati"
710 print"premi 'x' per terminare"
720 print"[Down]nota: durante la creazio
    ne del file"
725 print"premi <return> due volte per t
    erminare"
730 print"l'input"
740 call get-valid-key
750 return
```



Utility per il disk drive

Due ruotine per la sicurezza del software



Già in passato, su alcune pagine di questa rivista sono stati affrontati problemi riguardanti la protezione software. In particolare sono stati trattate alcune tecniche per salvaguardare singoli programmi o file dati, da maldestri tentativi di copiatura o da consumismi indesiderati. Le routine che presentiamo vogliono essere tali da completare la serie di utility per il drive, che sono state fino ad ora pubblicate. La prima, da applicare su file programmi, fornisce a questi l'opzione di auto-run e di resettamento del processo al termine dello stesso (cioè si ricomincia l'esecuzione); la seconda invece può essere applicata su singoli file dati o sull'intero dischetto: è molto efficace, come del resto la prima routine, per evitare possibili copiature e cancellazioni (quest'ultimo solo per la seconda opzione).

Assicurarsi con AutoRun

Chi desidera dotare il proprio programma di alcune forme di protezione, da quelle a livello di data entry a quelle sull'intero codice non ha che l'imbarazzo della scelta: per fare un esempio, tempo fa vi avevamo dato alcuni consigli per manipolare a vostro piacimento alcuni puntatori, all'interno del C64.

Nonostante ciò riteniamo che non sono sufficienti tutte le protezioni di questo mondo, se tuttavia si possono ancora usare LOAD e SAVE per impadronirsi di programmi,anche se protetti, con una facilità incredibile.

A questo punto, per evitare una temibile fase di SAVE dopo una di LOAD, la soluzione di tutti i problemi può essere risolta dall'applicazione di un autorun ai programmi che si vogliono proteggere.

Evitato quindi il pericolo di una possibile copiatura dopo il caricamento da dischetto, esso rimane comunque sempre in agguato una volta che il programma termina la sua esecuzione; se così fosse l'autorun non varrebbe chissà che come protezione. In realtà si è risolto anche questo inghippo, adottando, oltre all'autorun, pure l'auto-restore; cioè una volta terminato il processo, il C64 lo fa ricominciare, evitando così che il controllo della macchina si trasferisca alla tastiera e quindi nelle mani di malintenzionati pirati.

A completamento di tutto ciò si è anche disabilitato il tasto di RUN/STOP e quello di RESTORE per impedire uscite forzate dal processo. Ogni eventuale tentativo di break viene sistemato con un restore, da parte del C64.

Tecnicamente l'applicazione dell'autorun ad un programma, coincide con l'aggiunta di alcuni dati, ben nascosti, in testa a questo. Come tutti gli utenti Commodore devono sapere, i primi due byte di un programma rappresentano l'indirizzo di memoria dal quale la routine di LOAD deve iniziare a memorizzare i dati, prelevati dalle unità a dischi o nastri. La nostra routine modifica appunto questo indirizzo, facendolo puntare altrove.

La porzione di codice aggiunta in testa al programma ha quindi il compito di eseguire l'autostart una volta terminata la fase di caricamento. Risulta perciò arduo manipolare il programma, senza doversi imbattere in impedimenti quasi insormontabili.

Ma vediamo come si implementa e come si utilizza questa protezione. Dopo aver inserito in memoria la routine riportata nel primo listato, fatela eseguire: vi sarà chiesto il nome del programma (non file dati!) su cui applicare l'autorun; dopo alcuni controlli vi sarà chiesto il nome da dare alla versione protetta; dopodiché, se non vi sono irregolarità, la routine collega i dati relativi all'autorun. Questa fase richiede un certo tempo di attesa da parte vostra: un po' di pazienza, per favore.

Ad implementazione ultimata, provate a caricare il nuovo programma; se in esso non ci sono input vulnerabili o qualcos'altro che possa fargli sfuggire il controllo, allora il programma è da considerarsi ben protetto da copiatori. L'unico modo per riavere il controllo del C64 consiste nel resettare l'intero sistema.

Il caricamento può avvenire tramite un LOAD rilocabile o no: nel secondo caso i dati letti da disco vengono inseriti in memoria nelle medesime locazioni in cui erano collocati prima del SAVE.

Se pensate che oltre al fattore protezione sia da apprezzare l'utilità dell'autorun, allora dovete ricorrere al caricamento non rilocabile, cioè far uso del comando:

```
LOAD"Programma Protetto",8,1
```

In questo caso si ha l'autorun vero e proprio. Chi invece non conosce le mo-

dalità oppure chi è stato preso da istinti pirateschi non ha che da scegliere il classico LOAD, senza la clausola ',1'. Il caricamento avviene normalmente, ma poi stranamente appare il cursore: mah, che razza di protezione e'? Diamo un LIST e ..., sorpresa!

L'autorun, comunque, non risulta una protezione inespugnabile per un pirata esperto e collaudato; ma per un utente medio è certamente molto sicuro per scoraggiare ogni cattiva intenzione.

Le difese del drive

La seconda routine presentata si rivela particolarmente utile a chi possiede dati molto importanti su dischetto; di solito sono informazioni che vengono lette di frequente dall'utente, ma troppo preziose per rischiare di essere accidentalmente (o volontariamente) cancellate.

Quello che vi proponiamo è quindi una protezione anti-scratch irreversibile, da applicare a singoli file: questi potranno essere letti e copiati, ma non cancellati. Irreversibile significa che, una volta innestata la protezione su un file, questo non potrà essere tolto dal dischetto con le normali operazioni del DOS, a meno di formattazioni.

Essendo comunque possibile la copiatura, non si può considerare l'anti-scratch come una tecnica totale di protezione; ma come vedremo, la stessa routine offre altre due possibili soluzioni, una agente su singoli file e l'altra sull'intero dischetto, la prima integrabile con l'anti-scratch: com'è vero che l'unione fa la forza, anche in questo caso l'impiego di due tecniche assieme rende ancora più problematico il lavoro dei malintenzionati e ficcanaso.

L'implementazione dell'anti-scratch è semplice: appena fate eseguire la routine, vi sarà chiesto di scegliere se la protezione va applicata ad un singolo file oppure al dischetto; nel caso nostro è da scegliere la prima opzione. Dopodiché dovrete dichiarare quale delle due opzioni possibili scegliete (anche entrambe): per l'anti-scratch rispondete affermativamente.

Quindi verrà richiesto il nome del file da proteggere; pochi attimi di attesa ed è fatto! Per sincerarvi dell'avvenuta operazione, leggete la directory: accanto al nome del file interessato c'è il carattere '<' che simboleggia la presenza della protezione.

La seconda forma di protezione, per quanto riguarda un singolo file, consiste nel modificarne il numero di blocchi, quello per intenderci indicato prima del filename, quando leggiamo la directory; anche se a prima vista non si riescono a scorgere gli effettivi benefici, questo e' sicuramente un modo per creare un po' confusione.

Ricordate che nel gioco della protezione, la condizione di caos e di imprevedibilità che riuscite a creare attorno al vostro software gioca sempre a vostro favore. Talvolta la confusione può essere apprezzata come un prezioso alleato: Jovannotti? Per carità!

La terza ed ultima protezione disponibile nella routine è anche la più efficiente: riguarda l'intero dischetto, è di rapida implementazione e fa in modo che tutte le operazioni di scrittura su drive vengano disabilitate; ne segue che risulta impossibile cancellare i file presenti (anche programmi) e che anche le operazioni di backup vengono negate; tuttavia rimane sempre possibile leggere singoli file e copiarli su altri dischetti.

Fate attenzione che, se avete eseguito un'operazione illegale, come scrittura o cancellazione, anche la copiatura non è permessa, a meno di un reset del C64.

Tutto ciò è stato reso possibile modificando l'identificatore della versione del DOS (ID), presente sulla superficie di ogni dischetto. Nel riquadro sottostante ci sono tutti i dati utili per la procedura di protezione:

Drive	ID	Indice	Collocazione
1541	65	2	Traccia 18, Settore 0

L'ID del DOS 1541 è 65. Una modifica a questo valore all'interno del dischetto, alla traccia e settore indicati, con indice 2, cioè al terzo byte dall'inizio del settore, produrrà gli effetti desiderati.

Come già sapete, l'utilizzo di questa routine è piuttosto semplice: è sufficiente che voi specifichiate le vostre intenzioni; quindi dopo una minima attesa la protezione è già implementata.

Siccome la routine effettua delle scritture su singoli blocchi fisici del disco (cioè vengono specificati traccia e settore), allora vi consigliamo di utilizzare un dischetto di prova, finché non siete sicuri che la routine funzioni bene.

Prima di lasciarvi ai listati, vi diciamo che la seconda routine può essere migliorata, aggiungendo ulteriori 'precauzioni'; ad esempio si potrebbe fare in modo di poter cambiare i nomi dei file, modificare i puntatori relativi a successivi blocchi della directory oppure i puntatori al primo blocco di dati, collegato al file protetto. Sono tutti piccoli trucchi che, se presi assieme, formano un vero e proprio rompicapo, anche per i pirati più esperti.

Novità

Nella seconda routine vengono effettuate delle operazioni su disco che per molti di voi risultano nuove o quasi. Si trattano di istruzioni eseguite su file di tipo random, cioè un file nel quale si può accedere direttamente ai singoli record, specificando l'indirizzo fisico del settore (blocco fisico) a cui appartengono. È un tipo di programmazione senza dubbio avanzato, la quale richiede una notevole cura per evitare di rovinare involontariamente informazioni fondamentali per il dischetto.

Parlando di record fisico di un file random, ci riferiamo sempre al blocco fisico, formato da 256 byte; non deve essere confuso col record logico, che consiste invece in una serie di caratteri che compongono l'insieme dei campi: è per capirci quello identificato da un tracciato record.

In un file random sono ovviamente possibili la scrittura e la lettura; per operare su tali file bisogna far uso di due canali, il 15 (canale dei comandi) e uno a scelta fra 2 e 14, entrambi muniti di buffer.

L'istruzione PRINT# causa una sempre una scrittura su buffer; esiste una particolare istruzione inviata sul canale comandi (15), che permette di trasferire il contenuto del buffer (non quello comandi) in un determinato blocco fisico, specificandone l'indirizzo.

Le istruzioni INPUT# e GET# servono per leggere dal buffer; anche per la lettura esiste una istruzione che permette di trasferire nel buffer il contenuto di un blocco fisico del dischetto.

Per utilizzare i due canali, sono necessarie due operazioni di OPEN, e precisamente:

OPEN 15,8,15

per aprire il canale dei comandi.

OPEN nf,8,is,"#"

per aprire il canale dati; 'nf' e' il numero logico del file, mentre 'is' e' l'indirizzo secondario: entrambi possono assumere un valore compreso fra 2 e 14 e di solito per comodità sono uguali (non usate il 4, riservato alla stampante).

L'operazione di OPEN mette quindi a disposizione un canale ed un buffer (di 256 byte), ai quali si fa riferimento per mezzo del valore 'nf'.

Vediamo ora i comandi che possono essere lanciati nel canale comandi, con una PRINT#:

Comando	Abbreviazione
BLOCK-READ	B-R U1
BLOCK-WRITE	B-W U2
BLOCK-EXECUTE	B-E
BUFFER-POINTER	B-P
BLOCK-ALLOCATE	B-A

BLOCK-FREE	B-F
Memory-Write	M-W
Memory-Read	M-R
Memory-Execute	M-E

Per i primi sei comandi l'abbreviazione e' facoltativa. Per i rimanenti è invece obbligatoria.

La seconda routine presentata fa uso di alcuni di questi comandi. Vediamoli:

BUFFER-POINTER. Consente di spostare il puntatore all'interno del buffer specificato; la sua posizione è fondamentale, poiché le operazioni di PRINT#, INPUT# e GET# operano a partire da essa. Di solito si sposta il puntatore dopo aver trasferito nel buffer il contenuto di un blocco (con B-R o U1), oppure prima di scrivere in un blocco il contenuto del buffer (con B-W o U2).

PRINT#15,"B-P";nb;pos

Dove 'nb' e' il numero logico del canale e 'pos' la nuova posizione del puntatore nel buffer (fra 0 e 255). U1 (oppure UA). Trasferisce il blocco specificato nel buffer, questo assegnato al momento della OPEN. A differenza di B-R, U1 legge senza tenere conto della posizione finale del puntatore, all'interno del buffer (non si cura cioe' dell'EOF). Di solito si preferisce a B-R.

PRINT#15,"U1";nb;0;tr;set

Dove 'nb' e' lo stesso di prima, '0' indica il tipo di drive (1541), 'tr' la traccia e 'set'il blocco dai quali leggere. U2 (oppure UB). Trasferisce il contenuto del buffer nel blocco specificato. A differenza di B-W, U2 non registra la posizione finale del puntatore nel buffer. Di solito si preferisce a B-W.

PRINT#15,"U2";nb;0;tr;set

M-W (memory write). Consente di registrare nella RAM del disk drive, una quantita' di al massimo 34 byte. I parametri da passare sul canale 15 sono nell'ordine il byte basso e il byte alto dell'indirizzo di memoria dal quale iniziare la scrittura, numero di byte da trasferire e quindi i byte medesimi. M-R (memory read). Consente di leggere un byte appartenente alla RAM o ROM del disk drive, facendo uso del canale comandi. I parametri da passare sono il byte basso e quello alto dell'indirizzo di memoria dal quale leggere il byte. Tale indirizzo va spedito sottoforma di stringa, come segue:

PRINT#15,"M-R:"chr$(X)chr$(Y)

dove Y*256+X è appunto l'indirizzo. Questa istruzione preleva il byte dal dischetto e lo memorizza nel buffer; a questo punto per averlo, si deve operare con:

GET#15,D$

con D$ contenente il dato letto.

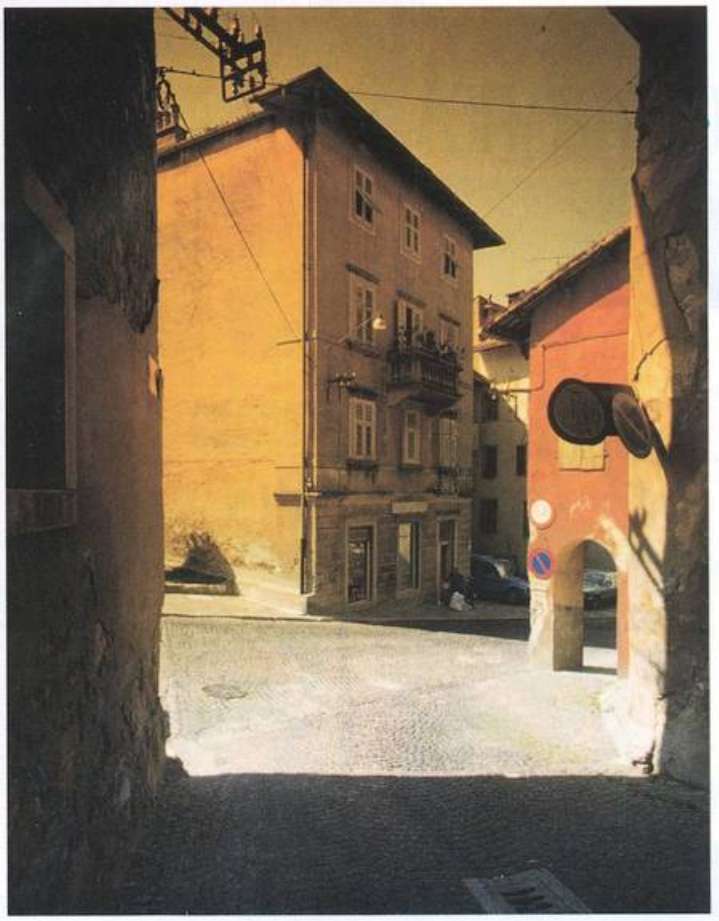
```
  **********
  ** AutoRun **
  **********

1 poke53281,0: poke53280,0
5 print"[Clr][Down] auto start 64":
  print
10 restore: open 15,8,15
20 input"[Ctrl+8]nome del programma[Ctrl
   +2]";n$
30 open 1,8,3,n$+",p,r":
   input#15,e,e$,e1,e2
40 if not e then get#1,a$,b$:
   if a$<>chr$(1) then e=1:
   e$="[Down]non esiste!"
50 if e then print e$: close 1: stop
60 print: input"[Ctrl+8]nome del program
   ma convertito[Ctrl+2]";c$
70 open 2,8,4,"0:"+c$+",p,w":
   input#15,e,e$,e1,e2
80 if e then print "[Down]gia' esiste!":
   close 2: stop
85 print: print" aspetta...";
90 data 192,  2,  0,  8,  1,  0,147, 34
100 for j=0 to 7: read x:
    print#2,chr$(x);: next j
110 for j=1 to len(c$):
    print#2,mid$(c$,j,1);: next j
120 data  34, 44, 56, 44, 49
130 for j=0 to 4: read x:
    print#2,chr$(x);: next j
140 for j=0 to 52-len(c$):
    print#2,chr$(0);: next j
150 data 139,227, 52,  3,124,165, 26,167
160 data 228,167,134,174,  0,  0,  0,  0
170 data  76, 72,178,  0, 49,234,102,  2
180 data  71,254, 74,243,145,242, 14,242
190 data  80,242, 51,243, 87,241,202,241
200 data 237,246, 62,241, 47,243,102,254
210 data 165,244,237,245, 32, 89,166, 76
    ,174,167
220 for j=0 to 57: read x:
    print#2,chr$(x);: next j
230 for j=0 to 1221: print#2,chr$(32);:
    next j : print#2,chr$(0);
240 get#1,a$: sw=st:
    if len(a$)=0 then a$=chr$(0)
250 print#2,a$;: if sw=0 goto 240
260 close1: close 2: close 15
265 print"fatto!": print
270 print"altri programmi [s/n]";a$
280 get a$: if a$="" then 280
285 if a$="s" then 5
290 sys peek(65532)+peek(65533)*256

    **********
    ** anti-scratch **
    **********

100 rem ** protezione per drive **
110 rem *- protezione antiscratch file
120 rem *- cambio del numero di blocchi
130 rem *- protezione del dischetto
145 cs$=chr$(147): dc$=chr$(17):
    ry$=chr$(18): rn$=chr$(146)
150 print cs$"protezione per drive"
160 rem * parametri drive 1541
170 dt=18: dl=144: dh=2: di=4: rv=1:
    pt=18: ps=0
210 print dc$"protezione: (1) file -(2)
    disco";
220 input fd$: if fd$="2" then 475: rem
    * protezione del disco
230 input"filename:";f$
235 if len(f$)>16 then 230
240 input"per evitare cancellazioni (s/n
    )";sp$: sp=sp$="s"
```

```
245 input"cambio del numero di blocchi (
    s/n)";cb$: cb=cb$="s"
250 if cb=0 then 280
255 input"nuovo numero di blocchi";nb
260 if nb>65535 then 255
265 ch=int(nb/256): cl=nb-ch*256
270 :
275 rem * controllo esistenza file
276 rem * continua se esiste
280 open 15,8,15
285 open 5,8,5,f$: get#5,a$:
    if st then 460
295 print#15,"m-r"chr$(dl)chr$(dh);: rem
    *trova il settore del file nella dir
300 get#15,s$: sec=asc(s$+chr$(0))
305 print#15,"m-r"chr$(dl+di)chr$(dh);:
    rem trova l'indice nella directory
310 get#15,i$: ind=asc(i$+chr$(0))
315 :
320 open 6,8,6,"#"
325 print#15,"u1";6;val(d$);dt;sec;: rem
    * abilita la lettura dei dati
330 print#15,"b-p";6;0;: rem * posiziona
    il buffer all'inizio
335 get#6,pl$: if pl$="" then pl$=chr$(0
    ): rem * blocco basso della dir
340 get#6,ph$: if ph$="" then ph$=chr$(0
    ): rem * blocco alto della dir
345 if ind=2 then 365:  rem * e' il
    primo file inserito
350 for x=2 to ind-1: get#6,a$: ss$=ss$+
    chr$(asc(a$+chr$(0))): next x
355 :
360 rem *tempo per introdurre i dati
    effettivi del file scelto
365 get#6,ty$: if ty$="" then ty$=chr$(0
    ): rem * tipo di file
370 get#6,tr$: if tr$="" then tr$=chr$(0
    ): rem * traccia del primo dato
375 get#6,se$: if se$="" then se$=chr$(0
    ): rem * settore del primo dato
380 for x=1 to 16: get#6,a$: na$=na$+
    chr$(asc(a$+chr$(0))): next x
385 for x=1 to 9: get#6,a$: bs$=bs$+
    chr$(asc(a$+chr$(0))): next x
390 get#6,bl$: if bl$="" then bl$=chr$(0
    ): rem * contatore di blocco basso
395 get#6,bh$: if bh$="" then bh$=chr$(0
    ): rem * contatore di blocco alto
405 if sp then ty$=chr$(asc(ty$) or 64):
    rem *set del bit 6 per l'antiscratch
410 if cb then bl$=chr$(cl):
    bh$=chr$(ch): rem * cambio del
    contatote di blocco
415 :
420 rem * inserisce il resto dei dati
425 if ind=226 then 440: rem * ci sono
    gia' dati
430 for x=ind+30 to 255: get#6,a$:
    es$=es$+chr$(asc(a$+chr$(0))): nextx
440 open 7,8,7,"#"
445 print#15,"b-p";7;0;: rem * posiziona
    il buffer all'inizio
450 print#7,pl$;ph$;ss$;ty$;tr$;se$;na$;
    bs$;bl$;bh$;es$;
455 print#15,"u2";7;val(d$);dt;sec;: rem
    * fa sapere dove si deve scrivere
460 close 5: close 6: close 7: close 15
465 print chr$(147)"protezione file comp
    letata!!": end
470 :
475 print chr$(147)"protezione del disco
    "
476 print"per non correre il pericolo di
     copiature"
485 open 5,8,5,"#": open 6,8,6,"#":
    open 15,8,15
490 for lp=1 to rv
495 print: print"passo"lp":traccia"pt"se
    ttore"ps
500 print#15,"u1";5;0;pt;ps:
    print#15,"b-p";5,0: rem * lettura
    dal ch#5
505 print#15,"b-p";6,0: rem * scrittura
    dal ch#6
510 b$="": for x=0 to 255
520 :get#5,a$: a$=left$(a$+chr$(0),1):
    if x=2 then a$="*"
525 :b$=b$+a$: if x=128 then c$=b$:
    b$=""
530 next x
535 print#6,c$;b$;:
    print#15,"u2";6;0,pt,ps
540 ps=ps+3: next lp
545 print#15,"i"+mid$(str$(0),2)+""
550 close 5: close 6: close 15
560 printchr$(147)"protezione del disco
```



INFORMATICA OGGI
SETTIMANALE 60
Anche le reti locali nel mirino di Olivetti
Wyse Technology obiettivo Italia
MECCANICA OGGI
SETTIMANALE 1
SETTIMA
ELETTRONICA · AUTOMAZIONE
INFORMATICA
BIT
SPECIALE UNIX
VETRINA
elettronica OGGI
STRUMENTAZIONE &MISURE OGGI
PC Floppy
PC MAGAZINE
Summa, gestionale per tutti
PC WORLD
MAGAZINE
Amiga Transactor
Trasmissione Dati e Telecomunicazioni
SUPER COMMODORE 64 e 128
CON DISCO
ROOT RACE
SKETCH PAD PLUS
OIL DEFENSE
ML CLONER
BASICALLY MUSIC
COMPUTER GRAFICA
& APPLICAZIONI
AMIGA MAGAZINE
CONTIENE DUE FLOPPY DISK CON PROGRAMMI
Compu Scuola
La rivista di informatica nella didattica per la scuola italiana
NUMERO SPECIALE
dedicato agli Atti di SCUOLA 2000
olivetti PRODEST
USER
LA PRIMA E UNICA RIVISTA INDIPENDENTE PER GLI UTENTI PC 128-PC 128 S-PC 1
PC 1
Peano, peano, come te non c'è nessuno
Xtree
PC 128 S
Sistema musicale
Le risposte di Eliza
PC 128
Sistemi di numerazione
Quadrato cinese
IL L.M. DEL PC 128
PC GAMES
IN QUESTO NUMERO:
CONCENTRATION • FREESTYLIN
TIC TAC TOE • ULTIME NOVITA'
CALCIO INDOOR
Tachimetro per bicicletta
Audiometro
Controller per impianti di riscaldamento
Voltmetro digitale per MSX
Accordatore d'antenna

ABBONAMENTO JACKSON = SERVIZIO COMPLETO

Da quest'anno l'abbonamento alle riviste Jackson offre una serie innegabile di vantaggi e servizi: anzitutto lo sconto eccezionale del 40% sul prezzo di copertina, pressochè doppio rispetto al passato, che Jackson ha voluto proporre ai lettori

per celebrare il decimo anno di attività. Inoltre, abbonarsi a Jackson garantisce l'accesso a una rete multinazionale di informazioni, grazie al recente accordo azionario con la VNU Business Press Group, maggiore editore tecnico internazionale del settore. Ma c'è di più: **la Jackson Gold Card,** per l'identificazione immediata del codice abbonamento, sarà recapitata gratuitamente agli abbonati e permetterà al titolare di usufruire di molteplici servizi gratuiti quali: sconto del 20% fino al 28/2/1989 e del 10% dopo tale data, sul prezzo di copertina di libri e software Jackson, per acquisti effettuati direttamente dall'editore, oltre a una serie di sconti per acquisti vari presso librerie, computershop e altri esercizi convenzionati in tutta Italia.

In più, il titolare di **Jackson Gold Card** potrà ottenere sconti sui corsi di formazione della Jackson S.A.T.A., la scuola Jackson di Alte Tecnologie Applicate, oltre all'abbonamento gratuito a 6 numeri di uno (a scelta) dei tre settimanali Jackson: "E.O. News Settimanale di Elettronica", "Informatica Oggi Settimanale" o il nuovissimo "Meccanica Oggi", annunciato per l'inizio del 1989.

Infine, l'abbonato ha diritto all'invio personalizzato e riservato dei cataloghi libri e della nuova rivista **"Jackson Preview Magazine",** con l'annuncio di tutte le novità editoriali Jackson.

GRUPPO EDITORIALE JACKSON

PRIMO NELLA BUSINESS-TO-BUSINESS COMMUNICATION

1° PREMIO

HONG KONG • BANGKOK • SINGAPORE

UN FANTASTICO VIAGGIO
IN ESTREMO ORIENTE
IN COLLABORAZIONE CON:

CENTRO TURISMO

swissair

Sheraton

I LEADER PER UN VIAGGIO DI SUCCESSO

2° PREMIO

UN COMPUTER
AMIGA 2000

6

UN COMPUT

dal 7° al 10° PREMIO

4 PACCHETTI SOFTWARE
"Commodore Software by CTO"

PREMIO
"NUOVO C64"

ABBONAMENTO JACKSON = FORTUNA STREPITOSA

AUT. MIN. RICH.

Abbonarsi alle riviste Jackson significa leggere il meglio, risparmiando il 40%, in informatica, elettronica e nuove tecnologie, ma soprattutto partecipare al grande concorso Jackson riservato agli abbonati, con la possibilità di vincere premi favolosi.

REGOLAMENTO DEL CONCORSO

1 – Il Gruppo Editoriale Jackson S.p.A. promuove un concorso a premi in occasione della Campagna Abbonamenti 1988/1989.
2 – Per partecipare è sufficiente sottoscrivere, entro il 31.3.1989, un abbonamento a una delle 30 riviste Jackson.
3 – Sono previsti 10 favolosi premi da sorteggiare fra tutti gli abbonati.
4 – Primo premio: un viaggio per due persone in Estremo Oriente, che prevede: passaggi aerei Swissair, pernottamenti in Hong Kong, Bangkok e Singapore, presso gli hotel: Royal Orchid Sheraton e Sheraton Towers della catena Sheraton Hotel, nonchè escursioni in luogo nelle tre suddette località.
Gli altri nove premi consistono rispettivamente (in ordine di esposizione) in:
1 computer Amiga 2000 completo di unità centrale con 1 MB di memoria, dischetto da 3" 1/2, tastiera, mouse, sistema operativo e monitor a colori 1084.
1 personal computer PC 20 III SERIE completo di unità centrale con 640 KB di memoria, dischetto da 5" 1/4, hard disk da 20 MB, mouse 1352, sistema operativo MS-DOS 3.20 monitor monocromatico e tastiera.
1 personal computer PC1 completo di unità centrale con memoria 512 KB, dischetto da 5" 1/4, tastiera, monitor monocromatico, sistema operativo MS-DOS 3.20 e GW-Basic.
1 computer Amiga 500 con 512 KB Ram e 256 KB Rom di memoria, sistema operativo e monitor a colori 1084.
1 computer "nuovo C64" completo di manuali e sistema operativo.
Dal settimo al decimo premio incluso, n. 4 pacchetti "Commodore Software by CTO".
5 – Gli abbonati a più di una rivista avranno diritto, per l'estrazione, all'inserimento del proprio nominativo tante volte quante sono le testate sottoscritte.
6 – L'estrazione dei 10 premi in palio avverrà presso la sede del Gruppo Editoriale Jackson entro il 30.5.1989.
7 – L'elenco dei vincitori, ad estrazione avvenuta, pubblicato su almeno 10 delle riviste Jackson. La vincita inoltre sarà pubblicata con lettera raccomandata a ciascuno dei sorteggiati.
8 – I premi verranno messi a disposizione degli aventi diritto entro 30 giorni dalla data dell'estrazione, ad esclusione del primo premio, il quale dovrà essere effettuato, compatibilmente con la disponibilità dei posti, in un periodo da definirsi, entro il 31.12.1989.
9 – Le spese di vitto relative al viaggio, nonchè l'eventuale controllo di manutenzione extra garanzia per i personal computer Commodore, saranno a carico dei rispettivi vincitori.
10 – I dipendenti, i familiari, i collaboratori del Gruppo Editoriale Jackson sono esclusi dal concorso.

Sempre quest'anno, il concorso abbonamenti Jackson prevede un primo premio veramente eccezionale: la possibilità di esplorare il misterioso Estremo Oriente, in un viaggio che unisce il fascino di una tradizione millenaria ad uno sviluppo tecnologico senza precedenti.

Il viaggio, di oltre dieci giorni per due persone, è studiato nei minimi dettagli, per offrire il miglior comfort e le migliori ospitalità ed è garantito da tre leader di primissimo livello: Acentro Turismo di Milano, Swissair e Sheraton Hotels.

Non solo. Ad altri nove abbonati fortunati, il Gruppo Editoriale Jackson, in collaborazione con Commodore Computer e CTO, riserva altri premi eccezionali, dalla più completa gamma di computer di successo: un favoloso personal computer Amiga 2000, un Commodore PC20 III serie, un Commodore PC1, un Amiga 500 e un nuovo C64, in palio dal secondo al sesto estratto. Quattro pacchetti "Commodore Software by CTO" saranno inoltre sorteggiati dal settimo al decimo premio.

Partecipare al concorso è semplice: basta abbonarsi a una o più tra le riviste Jackson (chi si abbona a più riviste ha, naturalmente, più possibilità di vincita), utilizzando la speciale Cartolina/Questionario, già predisposta e affrancata, da compilare in ogni sua parte e restituire all'editore.
Affrettatevi! Abbonatevi per vincere!

Commodore
LEADER IN PERSONAL COMPUTER

GRUPPO EDITORIALE JACKSON

1 PRIMO NELLA BUSINESS-TO-BUSINESS COMMUNICATION

ABBONAMENTO JACKSON = RISPARMIO ECCEZIONALE

Abbonarsi è semplice: basta compilare in ogni sua voce la speciale Cartolina/Questionario già predisposta e affrancata e rispedirla all'editore. Per il versamento dell'importo dell'abbonamento, utilizzate, preferibilmente l'apposito modulo di C.C.P. già predisposto e allegato alla rivista.

SERVIZIO QUALIFICAZIONE LETTORI

SPECIALE: PER CHI ACQUISTA LE RIVISTE JACKSON IN EDICOLA

Da quest'anno il Gruppo Editoriale Jackson ha predisposto uno **Speciale Servizio di Qualificazione Lettori e Abbonati,** che prevede l'assegnazione di una serie di dati relativi agli interessi specifici di ognuno, per poter offrire un servizio adeguato alle reali esperienze di aggiornamento del lettore.

Tutti i lettori interessati allo **Speciale Servizio Qualificazione Lettori,** e quindi anche i non abbonati, devono restituire, compilata nella parte **Qualificazione lettori,** la **Cartolina Questionario** già predisposta e affrancata.

Per chi la spedirà, il Gruppo Editoriale Jackson garantisce fin d'ora **GRATUITAMENTE:**

- Jackson Silver Card, che offre tutti i vantaggi della Gold Card, esclusi gli sconti sui libri riservati agli abbonati.

- Invio gratuito del **Catalogo Generale Libri Jackson.**
- Invio gratuito della **Jackson Preview Magazine.**
- **Abbonamento gratuito a sei numeri,** a scelta tra le seguenti riviste settimanali: E.O. News Settimanale - Informatica Oggi Settimanale - Meccanica Oggi (pubblicato da febbraio '89)

Area	Testate	Numeri Anno	Tariffa abbonam.	Tariffa intera
Elettronica e automazione	EO News Settimanale	40 + 6 omaggio	£. 59.500	£. 100.000
	Elettronica Oggi	20	£. 60.500	£. 100.000
	Automazione Oggi	20	£. 60.000	£. 100.000
	Meccanica Oggi	40 + 6 omaggio	£.59.000	£. 100.000
	Strumentazione e Misure Oggi	11	£ 39.000	£. 66.000
Informatica e Personal Computer	Informatica Oggi Settimanale	40 + 6 omaggio	£. 61.000	£. 100.000
	Informatica Oggi mese	11	£. 33.500	£. 55.000
	BIT (quindicinale da Gennaio)	20	£. 48.000	£. 80.000
	PC Magazine	11	£. 32.500	£. 55.000
	PC Floppy	11	£. 79.500	£. 132.000
	Computergrafica e applicazioni	11	£. 39.500	£. 66.000
	Trasmissione dati e Telec.	11	£. 34.000	£. 55.000
	Compuscuola	10	£. 24.500	£. 40.000
Tecnologie e mercati	WATT (quindicinale da Gennaio)	20	£. 36.500	£. 60.000
	LAB. NEWS	10	£. 30.000	£. 50.000
	Industria Oggi	11	£. 34.500	£. 55.000
	Media Production	11	£. 46.500	£. 77.000
	Strumenti musicali	11	£. 32.000	£. 55.000
Hobby e Home Computer	Fare Elettronica	12	£. 36.000	£. 60.000
	Amiga Magazine disk	11	£. 92.500	£. 154.000
	Amiga Transactor	6	£. 25.500	£. 42.000
	Commodore Professional 64/128 disk	11	£. 85.000	£. 143.000
	Commodore Professional 64/128 cass.	11	£. 59.500	£. 99.000
	Supercommodore 64/128 disk	11	£. 79.000	£. 132.000
	Supercommodore 64/128 cassetta	11	£. 49.500	£. 82.500
	Olivetti Prodest User	6	£. 18.000	£. 30.000
	PC Software	11	£. 66.000	£. 110.000
	PC Games 5 1/4"	11	£. 93.000	£. 154.000
	PC Games 3 ½	11	£. 99.500	£. 165.000
	3 ½ Software	11	£. 99.000	£. 165.000

Lo sconto del 40% è stato calcolato, in certi casi, arrotondando le cifre in modo da differenziare le tariffe di ciascuna rivista per esigenze di gestione.

GRUPPO EDITORIALE JACKSON

PRIMO NELLA BUSINESS-TO-BUSINESS COMMUNICATION

SCOPRI I JACKSON CENTER

Rivenditori specializzati nella vendita di manuali e testi di elettronica, informatica e comunicazioni.

BASILICATA

75100 **MATERA** - Planning - Piazza degli Olmi, 50 - Tel. 0835/263319

CALABRIA

88100 **CATANZARO** - C & G Computers - Via Acri, 26 - Tel. 0961/28076

CAMPANIA

80125 **NAPOLI** - Punto Quattro - Via Giulio Cesare, 21 - Tel. 081/634741 • 80134 **NAPOLI** - Top Electronics - Via S. Anna dei Lombardi, 12 - Tel. 081/5511115 • 84100 **SALERNO** - Computer Market - Corso V. Emanuele, 23 - Tel. 089/232051 • 84100 **SALERNO** - Infobit Shop - Via S. Leonardo, 120 - Tel. 089/335683

EMILIA ROMAGNA

47100 **FORLÌ** - Home e Personal Computer - Piazza Melozzo, 1 - Tel. 0543/35209 • 41100 **MODENA** - Viemme Autom. Ufficio - Via Emilia Est, 529 - Tel. 059/374037 • 43100 **PARMA** - Bit Show - Borgo Parente, 14/E - Tel. 0521/25014 • 42100 **REGGIO EMILIA** - Computerline - Via S. Rocco 10/C - Tel. 0522/32679

FRIULI VENEZIA GIULIA

34170 **GORIZIA** - B & S Elettronica Profess. - Viale XX Settembre, 37 - Tel. 0481/32193 • 34074 **MONFALCONE** (GO) - Tecnopower - Via S. Giacomo, 30 - Tel. 0481/44260 • 34122 **TRIESTE** - Computer Shop - Via P. Reti, 6 - Tel. 040/61602 • 33100 **UDINE** - Mofert - Viale Europa Unita, 41 - Tel. 0432/294620

LAZIO

03043 **CASSINO** (FR) - Computerline - Via Lombardia, 59 - Tel. 0776/277988 • 04023 **FORMIA** (LT) - A & R Elettronica - Via G. Paone, 1 - Tel. 0771/267876 • 00185 **ROMA** - S.I.S.CO.M. - I sottopassaggio Staz. Termini (ingr. metropolitana) - Tel. 06/4757798 • 00159 **ROMA** - Cartotib - Via Tiburtina, 614/D - Tel. 06/430808 • 00144 **ROMA** - Chopin - Via Chopin, 27 - Tel. 06/5916462 • 00192 **ROMA** - Computerline - Via Marcantonio Colonna, 10/12 - Tel. 06/384907 • 00199 **ROMA** - Computron Shop - Largo Forano, 7/8 - Tel. 06/8391556 • 00181 **ROMA** - R.T.R. - Via Gubbio, 44 - Tel. 06/7824204

LIGURIA

16121 **GENOVA** - ABM Computers - Piazza de Ferrari, 24/R - Tel. 010/296888 • 16139 **GENOVA** - Noxor - Via C. Centuriona, 1/4 - Tel. 010/317007 • 16154 **SESTRI PONENTE** (GE) - C.E.I.N. - Via Merano, 3/R - Tel. 010//673522 • 18039 **VENTIMIGLIA** (IM) - Computerlife B - Passeggiata Trento Trieste, 1 - Tel. 0184/299003

LOMBARDIA

24100 **BERGAMO** - Didatron - Via Moroni, 165 - Tel. 035/253092 • 24100 **BERGAMO** - Sandit - Via S. Francesco d'Assisi, 5 - Tel. 035/224130 • 21044 **CAVARIA CON PREMEZZO** (VA) - Curiotrè - Via Ronchetti, 71 - Tel. 0331/212585 • 20092 **CINISELLO B.** (MI) - G.B.C. Italiana - Viale Matteotti, 66 - Tel. 02/6181801 • 22100 **COMO** - Mantovani Tronic's - Via Caio Plinio, 11 - Tel. 031/263173 • 26100 **CREMONA** - Archimede - Via Palestro, 11/B - Tel. 0372/34545 • 22053 **LECCO** (CO) - Executive - Via Bovara, 16 - Tel. 0341/364706 • 20035 **LISSONE** (MI) - Computeam - Via Vecellio, 41 - Tel. 039/481010 • 20075 **LODI** (MI) - M.B.M. Informatica Systems - Corso Roma, 112 - Tel. 0371/53610 • 21016 **LUINO** (VA) - Hacker Studio - Via Veneto, 4/A - Tel. 0332/531126 • 46100 **MANTOVA** - Computer - Galleria Ferri, 7 Tel. 0376/325616 • 20154 **MILANO** - Computer Line - Via Maroncelli, 12 - Tel. 02/6552921 • 20124 **MILANO** - G.B.C. Italiana - Via Petrella, 6 - Tel. 02/203608 • 20144 **MILANO** - G.B.C. Italiana - Via Cantoni, 7 - Tel. 02/437478 • 20159 **MILANO** - Hex Electronic - Viale E. Jenner, 16 Tel. 02/6890898 • 20155 **MILANO** - Newel - Via Mac Mahon, 75 - Tel. 02/323492 • 20145 **MILANO** - Trend Electronics - Via Mascheroni, 14 - Tel. 02/437385 • 20052 **MONZA** (MI) - BIT 84 - Via Italia, 4 - Tel. 039/320813 • 20052 **MONZA** (MI) - C.S.I. Centro Studi Inf. - Via V. Emanuele, 24 - Tel. 039/325069 • 27100 **PAVIA** - Reo Elettronica - Via Briosco, 7 - Tel. 0832/473973 • 21018 **SESTO CALENDE** (VA) - J.A.C. Nuove Tecnologie - Via Matteotti, 38 - Tel. 0331/923134 • 20070 **SORDIO** (MI) - Tutto Software - Via Emilia, 22 - Tel. 02/9810339 • 21100 **VARESE** - Elettronica Ricci - Via Parenzo, 2 - Tel. 0332/281450

PIEMONTE

15100 **ALESSANDRIA** - Bit System - Via Savonarola, 13 - Tel. 0131/445692 • 15100 **ALESSANDRIA** - Campari Personal e Minicomputer - Corso Crimea, 63 - Tel. 0131/446826 • 13051 **BIELLA** (VC) - C.S.I. Teorema - Via Losana, 9 - Tel. 015/28622 • 13051 **BIELLA** (VC) - Informatica Biella - Piazza S. Paolo, 1 - Tel. 015/24181 • 10093 **COLLEGNO** (TO) - HI-FI Club - Corso Francia, 92/C - Tel. 011/4110256 • 12100 **CUNEO** - Rossi Computer - Corso Nizza, 42 - Tel. 0171/63143 • 10136 **TORINO** - Area Computer - Via Tripoli, 68 - Tel. 011/396669 • 10126 **TORINO** - Gruppo Sistemi Torino - Via Ormea, 83 - Tel. 011/6698114 • 10128 **TORINO** - Input Computer Studio - Corso Einaudi, 8 - Tel. 011/595594 • 15057 **TORTONA** (AL) - Karto 2000 - Via Emilia, 168 Int. - Tel. 0131/802215 • 28044 **VERBANIA INTRA** (NO) - I.G.S. - Corso Cobianchi, 5/7 - Tel. 0323/53660

PUGLIA

70125 **BARI** - Archimede - Viale Unità d'Italia, 32 - Tel. 080/227475 • 70051 **BARLETTA** (BA) - Aerre Computer - Via Indipendenza, 26 - Tel. 0883/301171 • 71100 **FOGGIA** - I.S.I. Informatica Sistemi - Via Matteotti, 83 - Tel. 0881/72823 • 74100 **TARANTO** - Elettrojolly Centro - Via De Cesare, 13 - Tel. 099/25534

SARDEGNA

09100 **CAGLIARI** - Computer Shop - Via Oristano, 12 - Tel. 070/653312 • 09100 **CAGLIARI** - INF. TEL. - Via Pergolesi 28/A - Tel. 070/491443 • 07026 **OLBIA** (SS) - Linea Ufficio - Via Galvani, 34 - Tel. 0789/57075 • 07100 **SASSARI** - Bajardo - Viale Italia, 16 - Tel. 079/233132

TOSCANA

50122 **FIRENZE** - S.I.T.T. - Borgo S. Croce, 11/R - Tel. 055/245892 • 57123 **LIVORNO** - Eta Beta Computer e Video - Via S. Francesco, 30 - Tel. 0586/886767 • 54100 **MASSA** - Bite Byte - Via Angelini, 19 - Tel. 0585/47785 • 52025 **MONTEVARCHI** (AR) - Tuttocomputer - Via Don Minzoni, 16 - Tel. 055/901504 • 56100 **PISA** - It Lab - Via Marche, 8/A - Tel. 050/552590

UMBRIA

05035 **NARNI** (TR) - Fortunati Ing. Giuseppe Comp. - Vicolo Torto, 2 - Tel. 0744/726993 • 06100 **PERUGIA** - Studio System - Via R. d'Andreotto, 49 - Tel. 075/757250 • 06049 **SPOLETO** (PG) - C.H.S. Computer's Home Spoleto - Viale Trento e Trieste, 67 - Tel. 0743/48029

VENETO

32100 **BELLUNO** - C.B.L. Computers - Piazza Mazzini, 15 - Tel. 0437/212204 • 35126 **PADOVA** - Computer Point - Via Roma, 63 - Tel. 049/22564 • 31100 **TREVISO** - E.L.B. Telecom - Via Montello, 13/A - Tel. 0422/66600 • 37122 **VERONA** - Personal Ware - Via Volto S. Luca, 6 - Tel. 045/592708 • 36100 **VICENZA** - Francomputer - Corso Fogazzaro, 139 - Tel. 0444/236669-542678 • 31029 **VITTORIO VENETO** (TV) - M.C.E. Elettronica - Viale V. Emanuele II, 56/D - Tel. 0438/555143

GEJ 0066

LA SOLUZIONE ALLE TUE ESIGENZE

Print Loader

Ecco una tecnica rivoluzionaria che piacerà agli amanti del Linguaggio Macchina e delle tecniche avanzate

I programmi Basic spesso utilizzano brevi routine in Linguaggio Macchina per leggere il disco, ordinare i vettori o realizzare una certa funzione più velocemente e più facilmente di come si potrebbe fare con il Basic.

Ma si fa spesso una fatica notevole per conciliare il Basic con le subroutine in L.M.

I metodi più comuni per unire le routine L.M. ai listati in Basic sono:

— caricare i blocchi di memoria contenenti i codici da disco (o da cassetta) servendosi del comando Basic LOAD"nome",#device,1

— attaccare i codici alla fine del Basic e salvare come un unico blocco di dati sia il programma Basic che i codici LM.

— usare i caricatori tipo READ/DATA/POKE. Ognuno di questi metodi ha una sua utilizzazione particolare e possiede pregi ma anche difetti.

La quarta tecnica

In questo articolo vi presentiamo un nuovo metodo (stabiliamo di chiamarlo "LM PRINT") che usa il comando Basic "PRINT" per scrivere in memoria i codici L.M.

Facciamo un esempio.

Prima di tutto, accendete il calcolatore, andate in modo maiuscolo/minuscolo e digitate la seguente riga esattamente come la trovate riportata. Se le lettere sono in maiuscolo, ricordatevi di digitarle premendo SHIFT. Ricordate che i simboli tra parentesi, sono in realtà comandi. Ad esempio dove si legge (com+b) dovete digitare assieme i tasti Commodore e "b". Dove si legge (SHIFT+(SPACE)), dovete premere contemporaneamente il tasto SHIFT e la barra spaziatrice.

```
20 ?"(RVS ON)(SPACE)(REVERSE
OFF)@(REVERSE ON)9(REVERSE
OFF)(COM+m)[(REVERSE
ON)(COM+a)(REVERSE OFF)g(SPACE)
(REVERSE ON)R(COM+b)H(REVERSE
OFF)LZ[(SHIFT+(SPACE)) (REVERSE
ON)s(REVERSE
OFF)QUESTI(SPACE)SONO(SPACE)I
(SPACE)DATI(SPACE)L.M.@'
```

Ora date RUN.

Dovrete vedere 40 caratteri sullo schermo, contando anche gli spazi. Controllate che ci siano tutti ed aggiungete la seguente riga:

```
10  POKE 209,88:POKE 210,27:POKE
211,0:POKE 213,45
```

Date di nuovo RUN.

Questa volta vedrete soltanto apparire il prompt del Basic dopo la scritta READY.

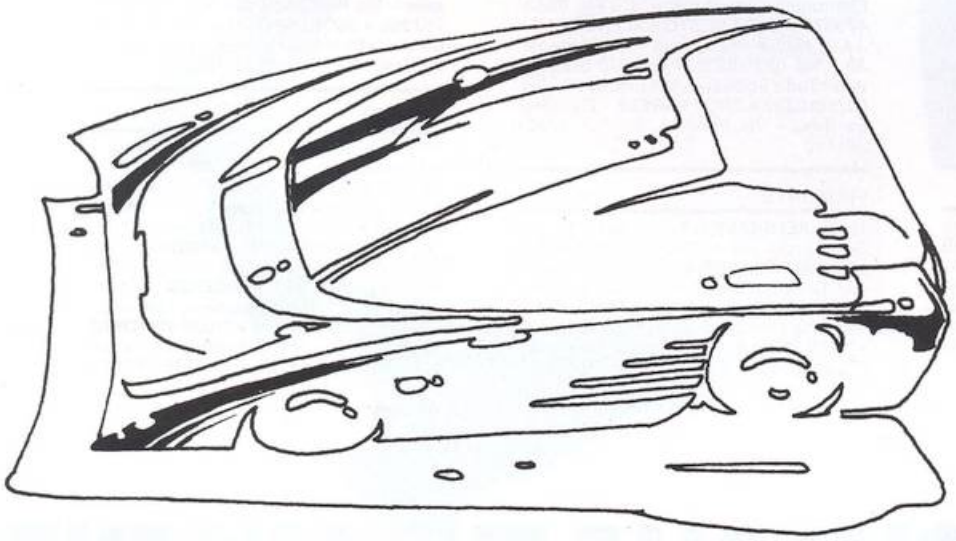
Il breve programma in Linguaggio Macchina sarà, però, già residente in memoria. Infatti, digitando SYS 7000 lo schermo sarà cancellato e verrà visualizzato il messaggio :

QUESTI SONO I DATI L.M.

Cerchiamo ora di capire qual è il meccanismo di funzionamento di questa nuova tecnica. Normalmente pensiamo al comando PRINT come ad un'istruzione del Basic riservata esclusivamente alla stampa dei caratteri sul video.

Sappiamo comunque, che il comando PRINT non fa altro che scrivere i codici ASCII dei caratteri in una specifica area di memoria, detta mappa video. Il circuito VIC II, responsabile tra l'altro della gestione degli sprite e della grafica del C64, provvederà a leggere e a visualizzare sul monitor i caratteri trovati nella mappa video.

Si potrebbe considerare il comando



```
10 rem*programma printloader*
20 print"[CLR][RVS-ON][CRSR-DOWN][CRSR-
   DOWN]printloader[RVS-OFF][CRSR-DOWN]
   [CRSR-DOWN]";chr$(13);"attendere..":
   cs=0
30 fori=25856to27832:readа:pokei,a
40 cs=cs+a:nexti
50 :
60 ifcs<>202679thenprint"**** errore
   nei data ****":end
70 print"il programma si attiva con:";
   chr$(13);"sys25856"
80 end
100 :
1000 data160,6,140,32,208,200
1010 data140,33,208,169,213,133
1020 data139,169,106,133,140,32
1030 data137,103,169,108,133,140
1040 data169,96,133,139,169,1
1050 data141,167,2,32,151,103
1060 data173,192,2,56,233,72
1070 data240,8,169,1,141,254
1080 data2,76,55,101,141,254
1090 data2,32,6,104,169,85
1100 data133,139,169,107,133,140
1110 data32,137,103,173,254,2
1120 data24,105,4,141,167,2
1130 data201,5,240,7,169,114
1140 data133,139,76,93,101,169
1150 data101,133,139,169,108,133
1160 data140,32,151,103,173,254
1170 data2,201,1,208,6,32
1180 data24,104,76,116,101,32
1190 data114,104,173,171,2,141
1200 data252,2,173,172,2,141
1210 data253,2,32,6,104,169
1220 data110,133,139,169,107,133
1230 data140,32,137,103,173,254
1240 data2,208,22,169,114,133
1250 data139,169,108,133,140,169
1260 data4,141,167,2,32,151
1270 data103,32,114,104,76,188
1280 data101,169,101,133,139,169
1290 data108,133,140,169,5,141
1300 data167,2,32,151,103,32
1310 data24,104,173,171,2,141
1320 data250,2,173,172,2,141
1330 data251,2,32,6,104,169
1340 data133,133,139,169,107,133
1350 data140,32,137,103,169,101
1360 data133,139,169,108,133,140
1370 data169,5,141,167,2,32
1380 data151,103,173,192,2,208
1390 data39,169,49,141,192,2
1400 data32,210,255,169,48,32
1410 data210,255,141,193,2,32
1420 data210,255,141,194,2,32
1430 data210,255,141,195,2,32
1440 data210,255,141,196,2,169
1450 data5,141,168,2,32,24
1460 data104,173,171,2,141,248
1470 data2,173,172,2,141,249
1480 data2,32,6,104,169,196
1490 data133,139,169,107,133,140
1500 data32,137,103,169,101,133
1510 data139,169,108,133,140,32
1520 data151,103,173,192,2,208
1530 data21,169,49,141,192,2
1540 data32,210,255,169,48,141
1550 data193,2,32,210,255,169
1560 data2,141,168,2,32,24
1570 data104,173,171,2,141,246
1580 data2,173,172,2,141,247
1590 data2,32,6,104,169,8
1600 data133,114,169,1,133,113
1610 data169,0,141,0,8,141
1620 data244,2,141,245,2,141
1630 data237,2,141,236,2,173
1640 data253,2,133,248,173,252
1650 data2,133,247,173,249,2
1660 data141,243,2,173,248,2
1670 data141,242,2,169,107,133
1680 data140,169,241,133,139,32
1690 data137,103,32,174,105,32
1700 data239,105,32,62,106,169
1710 data0,141,237,2,168,177
1720 data247,133,254,201,128,176
1730 data21,173,245,2,201,1
1740 data208,32,169,146,32,232
1750 data104,206,245,2,238,244
1760 data2,76,222,102,173,245
1770 data2,201,1,240,11,169
1780 data18,32,232,104,238,245
1790 data2,238,244,2,165,254
1800 data10,74,201,34,208,6
1810 data76,79,106,76,24,103
1820 data133,254,10,10,176,16
1830 data10,144,5,165,254,76
1840 data18,103,165,254,24,105
1850 data64,76,18,103,10,144
1860 data8,165,254,24,105,64
1870 data76,18,103,165,254,24
1880 data105,128,32,232,104,238
1890 data244,2,238,236,2,230
1900 data247,208,2,230,248,173
1910 data250,2,197,247,173,251
1920 data2,229,248,176,6,32
1930 data143,106,76,91,103,173
1940 data236,2,201,255,208,6
1950 data32,172,106,76,149,102
1960 data173,244,2,201,57,144
1970 data11,32,194,106,32,62
1980 data106,169,0,141,244,2
1990 data76,169,102,230,113,208
2000 data2,230,114,165,113,133
2010 data45,133,47,133,49,165
2020 data114,133,46,133,48,133
2030 data50,169,0,133,51,133
2040 data53,133,54,133,55,169
2050 data128,133,56,133,52,169
2060 data1,133,43,169,8,133
2070 data44,169,147,32,210,255
2080 data96,160,0,177,139,240
2090 data7,32,210,255,200,76
2100 data139,103,96,169,0,141
2110 data168,2,169,192,133,141
2120 data169,2,133,142,160,0
2130 data177,139,141,169,2,32
```

PRINT come un'istruzione di scrittura squisitamente sequenziale. Infatti se impartiamo un comando tipo:

PRINT "ciao"

otteniamo che i codici ASCII delle lettere "c,i,a,o" siano scritti nella mappa video, uno di seguito all'altro. Il passo sucessivo è ovvio: se sostituiamo al posto delle lettere del messaggio, dei codici ASCII corrispondenti a codici operativi di istruzioni in L.M., otterremo che, con semplici comandi PRINT, un nostro programmino in Linguaggio Macchina venga rapidamente trascritto nella mappa video. Il metodo appena descritto potrà sembrare un tantino limitativo. Prima di tutto perché la mappa video può contenere al massimo mille caratteri. Inoltre, così facendo, i nosti programmi in L.M.sarebbero esposti a possibili danneggiamenti dovuti alla normale gestione dello schermo, per esempio potrebbero essere spazzati via da una semplice pressione del tasto CLR/HOME.

Sappiamo però che, ogni volta che deve stampare qualcosa, il comando PRINT consulta l'attuale posizione del cursore sul video.

Questa informazione è contenuta in quattro locazioni della memoria. Difatti le locazioni 209 e 210 contengono rispettivamente la parte bassa e la parte alta dell'indirizzo dell'attuale linea del cursore, mentre nella locazione 211 viene scritta la posizione del puntatore sulla colonna. Nel nostro caso otterremo gli stessi risultati tenendo questa locazione sempre a zero ed agendo sull'indirizzo di linea del cursore.

Infine la locazione 231 indica al sistema operativo, (o meglio: all'interprete Basic) la lunghezza di una linea di schermo.

I valori normali sono 39 o 79 nel Commodore 64 e in genere devono rimane nel proprio range, pena strani effetti sul video. Ci proponiamo comunque di cambiare i contenuti di queste quattro locazioni a nostro vantaggio. Nella linea 10 dell'esempio che abbiamo visto, con le prime tre POKE si sposta il puntatore dello schermo alla locazione 7000. Con l'ultima POKE si informa l'interprete Basic che la lunghezza della linea è pari a 45 caratteri.

Avremmo potuto usare anche un numero più grande, fino ad un massimo di 255, comunque di parecchio più lungo dell'effettivo numero di codici che vogliamo memorizzare. Con la riga successiva, si è ottenuto che il Basic, non sapendo che nella locazione 7000 non c'è, in realtà, alcuna mappa video, scrivesse i nostri codici nel posto voluto.

Dopo aver dato SYS 7000 ed aver verificato il funzionamento del programmino, si può notare che ci viene mostrato nuovamente anche il normale prompt del Basic. Questo significa che i vettori del puntatore vengono ripristinati al loro valore originario alla fine di ogni comando PRINT. A causa di ciò, se dobbiamo maneggiare programmi in Linguaggio Macchina composti da grossi blocchi di codici, non possiamo far altro che usare una manciata di istruzioni PRINT terminanti con il punto e virgola e, ogni 255 byte di codici, ripetere l'alterazione del puntatore agendo sulle locazioni 209-210.

Alcuni problemi

Prima di procedere nella descrizione di questo particolare metodo è bene precisare alcune caratteristiche del comando PRINT. Per prima cosa, questa istruzione scrive in memoria i "codici schermo" dei caratteri e non il loro ASCII. Il manuale dell'utente del C64 riporta in appendice una tabella sia dei codici ASCII che dei codici schermo. Se andate a vedere vi accorgerete che i codici schermo sono solo 128. Inoltre, benché la sequenza alfabetica sia rispettata, molti simboli hanno posizioni diverse nelle due tabelle.

Se io digito: PRINT "A" (codice ASCII=65), in realtà il Basic memorizzerà il codice 1 perché l'1 corrisponde alla "A" nella tabella dei codici schermo. Quindi, se desidero memorizzare effettivamente il valore "65", dovrò inserire nell'istruzione PRINT uno strano carattere grafico.

Un altro problema sorge qualora volessimo memorizzare il codice numero 34. Infatti, a questo numero corrispondono le virgolette. Se non prendessimo particolari precauzioni, il Basic, incontrando le virgolette penserebbe che il testo da stampare sia terminato e, magari, interpreterebbe i sucessivi caratteri come nomi di variabili con conseguenze che vi lasciamo immaginare.

Siccome è così dificile generare correttamente le istruzioni di PRINT, ci si potrebbe chiedere se ci sono realmente dei vantaggi ad usare questo metodo piuttosto che altri. La risposta è affermativa, specialmente se si usa un programma per creare automaticamente le PRINT richieste!

Tra gli altri metodi citati all'inizio dell'articolo, il più diffuso è quello che si serve di istruzioni READ e DATA. Effettivamente questo metodo ha i suoi pregi.

Ad esempio, nel caso di listati corti, il risparmio di tempo è maggiore rispetto ad un caricamento separato delle routine. Inoltre, si evita di memorizzare separatamente la routine in L.M. sullo stesso nastro o disco. Per ultimo, è l'unico metodo accettabile (per motivi di chiarezza) nella pubblicazione del listato su una rivista. L'altra faccia della medaglia è, però, rappresentata dall'eccessiva lentezza e dal maggior spreco di memoria. Anche memorizzare alla fine del Basic i segmenti di memoria contenenti i codici L.M. può avere i propri vantaggi . Per mezzo di una manciata di POKE si può ingannare il Basic facendogli salvare su supporto magnetico sia il listato vero e proprio che i codici in Linguaggio Macchina. Se però si eseguono modifiche al listato o si aggiungono nuove righe si finisce addosso alla routine L.M

Vantaggi

Vediamo ora quali sono allora i vantaggi nell'utilizzare il metodo LM PRINT.

LM PRINT è dalle 10 alle 30 volte più

veloce dei metodi basati su READ/DATA/POKE grazie alla rapidità dell'istruzione di stampa del Basic. PRINT LM risparmia più della metà di RAM rispetto ad un sistema che memorizza le istruzioni in comandi DATA (dipende però dalla routine stessa).

Il programma

Il listato, presentato con questo articolo, si chiama "PRINT LOADER" e, se usato correttamente, vi può dare parecchie soddisfazioni.

Digitatelo e salvatelo. Una volta dato RUN aspettate qualche secondo per il caricamento dell'utility in memoria. Riottenuto il controllo del cursore, digitate:

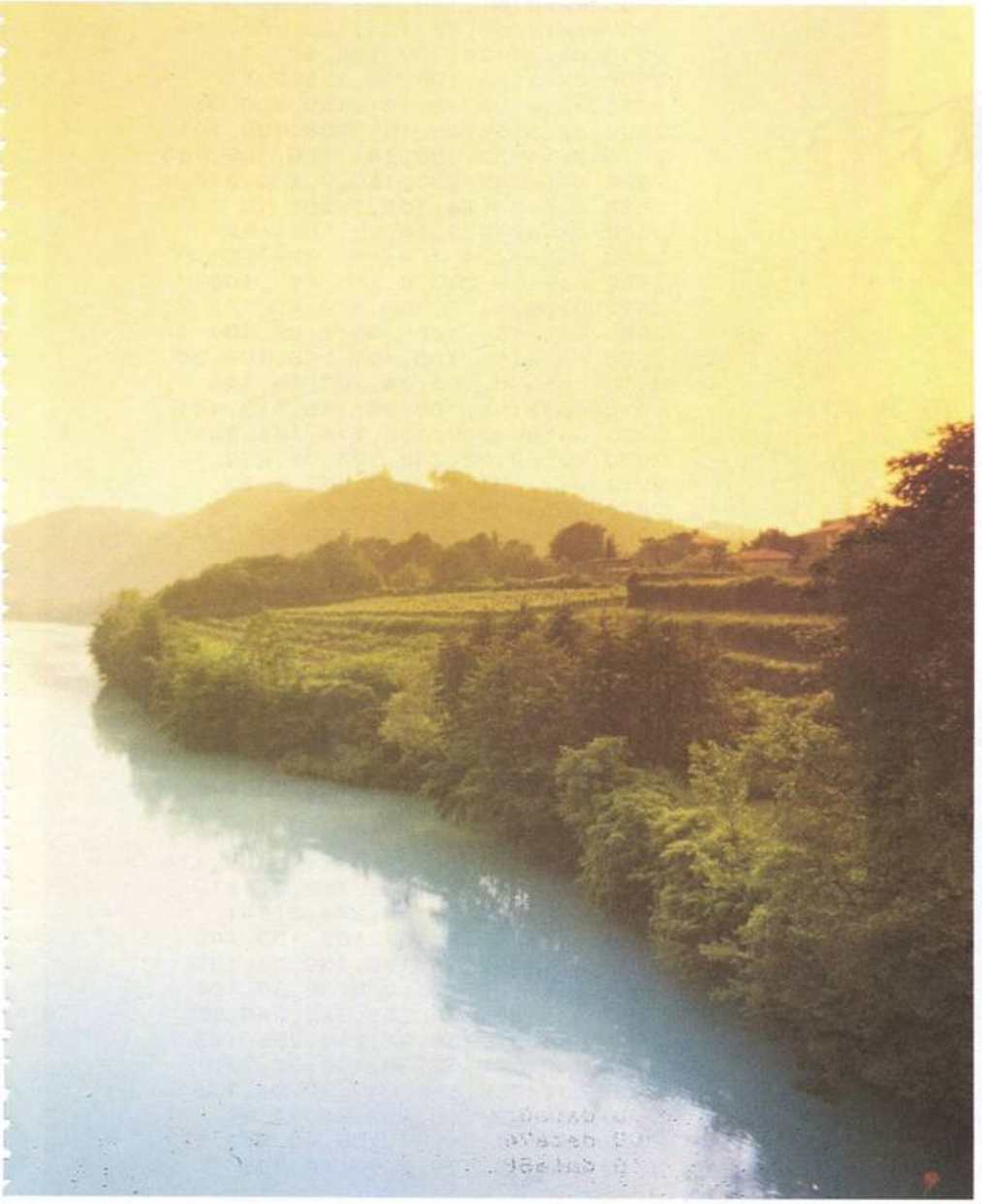
SYS 25856

L'utility vi chiederà se volete introdurre gli indirizzi in esadecimale (H) o decimale (D). Vi verranno chieste anche le informazioni necessarie per individuare la zona di memoria che volete convertire in istruzioni di PRINT.

A conversione finita verrà fatto ritorno al Basic dove, con un LIST potrete rendervi conto dell'operazione effettuata. Le linee di listato così generate possono essere salvate ed aggiunte ad un listato Basic con l'ausilio di un'utility tipo "Merge" o "Append".

Se non la possedete, aggiungete le righe di programma direttamente da tastiera, dopo aver generato quelle del programma in L.M.

Per facilitarvi il lavoro di aggiunta, PRINT LOADER vi chiederà in che modo generare i numeri di riga (inizio e incremento).



Se volete usare PRINT LOADER con maggiore facilità salvatene su disco la versione in puro Linguaggio Macchina. Lo potete fare con l'aiuto di un monitor (anche quello in Basic pubblicato a puntate su questa stessa rivista). Se non possedete un monitor, caricate in memoria PRINT LOADER, date RUN e, riottenuto il cursore del Basic, digitate:

CLR (RETURN)
POKE 43,0:POKE 44,101:POKE 45,185:POKE 46,108(RETURN)

Ora salvatelo su disco o su cassetta come fareste per un normale programma Basic, usando però un nome diverso dalla versione originale. Ad esempio :

SAVE"LMLOADER",8 (per il disco)

Spegnete poi la macchina o resettatela con SYS 64738 e provate a ricaricare il programma ricordandovi di aggiungere ",1" in coda all'istruzione di LOAD. Seguendo l'esempio di prima:

LOAD"LMLOADER",8,1

Un'avvertenza.

Se create programmi che fanno in memoria il PRINT di 1025 o più byte, potreste avere dei problemi con la tastiera durante il funzionamento del programma.

Apparentemente questo accade perché l'integrato CIA sembra confondersi se la lunghezza del fittizio schermo è maggiore di 1024.

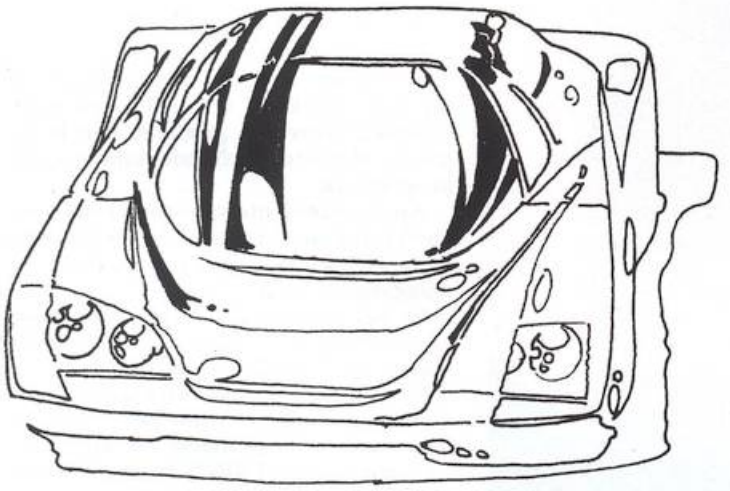
Se nel programma il controllo passa alle routine L.M. con una SYS subito dopo aver eseguito le istruzioni di PRINT, non si avranno malfunzionamenti. In caso contrario per eliminare il problema, aggiungete subito dopo l'ultima riga di PRINT, la seguente linea Basic:

SYS 64931:PRINT"[clr]"

Questo comando di pulizia-schermo rimetterà le cose a posto.

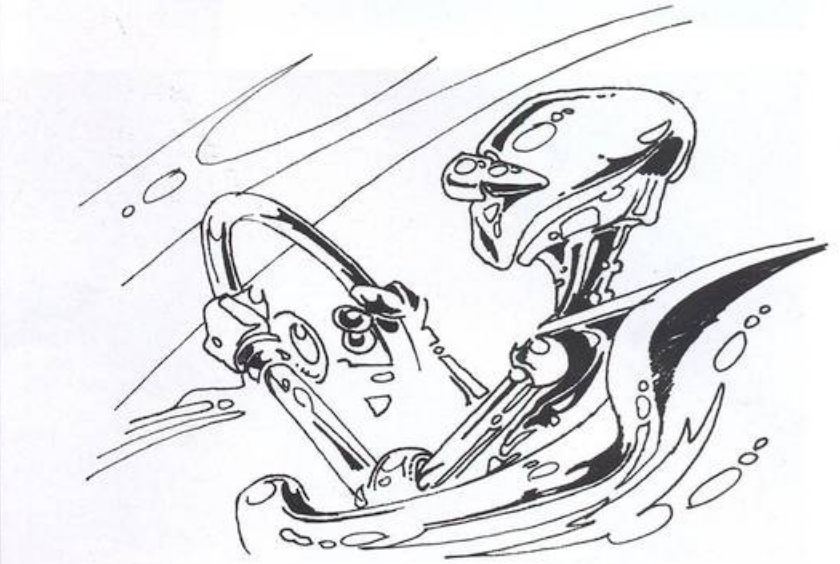
Un'ultima nota: PRINT LM è un metodo ideale per gestire al massimo una decina di Kilobyte. Ricordate che l'utility è allocata dall'indirizzo 25856. Se richiedete la conversione di blocchi di memoria più grandi di 8/10 K rischiate che, generando un gran numero di righe Basic, il programma si "scriva" addosso autodistruggendosi. Questa eventualità è più o meno probabile a seconda del fatto che stiate convertendo una zona di memoria "difficile". Se, ad esempio, l'utility incontra molti byte di valore superiore a 128 (o pari a 34), per convertirli dovrà generare righe Basic più lunghe della norma. In conclusione, se avete intenzione di unire al Basic routine in L.M. più lunghe di una decina di Kilobyte, usate uno dei metodi alternativi che abbiamo menzionato durante l'articolo.

```
2140 data159,255,32,228,255,172
2150 data169,2,209,139,240,6
2160 data136,240,233,76,180,103
2170 data201,20,240,54,201,13
2180 data208,15,168,165,197,201
2190 data64,208,250,162,0,202
2200 data138,208,252,152,96,172
2210 data168,2,170,173,167,2
2220 data205,168,2,240,195,138
2230 data32,210,255,145,141,238
2240 data168,2,165,197,201,64
2250 data240,250,160,0,136,208
2260 data253,76,164,103,172,168
2270 data2,240,167,206,168,2
2280 data32,210,255,76,234,103
2290 data160,15,169,2,133,142
2300 data169,192,133,141,169,0
2310 data145,141,136,16,251,96
2320 data173,168,2,170,169,0
2330 data141,170,2,141,171,2
```



```
2340 data141,172,2,169,192,133
2350 data141,169,2,133,142,172
2360 data170,2,177,141,72,14
2370 data171,2,46,172,2,173
2380 data171,2,172,172,2,14
2390 data171,2,46,172,2,14
2400 data171,2,46,172,2,24
2410 data109,171,2,141,171,2
2420 data152,109,172,2,141,172
2430 data2,104,56,233,48,24
2440 data109,171,2,141,171,2
2450 data144,3,238,172,2,238
2460 data170,2,202,208,190,96
2470 data173,195,2,208,26,173
2480 data194,2,141,195,2,173
2490 data193,2,141,194,2,173
2500 data192,2,141,193,2,169
2510 data48,141,192,2,76,114
2520 data104,169,0,141,170,2
2530 data169,192,133,141,169,2
2540 data133,142,160,0,177,141
2550 data141,174,2,238,170,2
2560 data172,170,2,177,141,168
2570 data173,174,2,72,152,32
2580 data222,104,141,173,2,104
2590 data32,222,104,10,10,10
2600 data10,13,173,2,168,173
2610 data170,2,201,3,240,13
2620 data152,141,172,2,238,170
2630 data2,172,170,2,76,160
2640 data104,152,141,171,2,96
2650 data56,233,48,201,16,144
2660 data2,233,7,96,162,0
2670 data129,113,230,113,208,2
2680 data230,114,96,169,48,141
2690 data165,108,141,166,108,141
2700 data167,108,166,248,232,169
2710 data0,202,240,43,24,105
2720 data1,201,10,240,3,76
2730 data3,105,169,1,24,109
2740 data166,108,201,58,240,6
2750 data141,166,108,76,1,105
2760 data169,48,141,166,108,169
2770 data1,24,109,165,108,141
2780 data165,108,76,1,105,24
2790 data105,48,141,167,108,96
2800 data169,48,141,156,108,141
2810 data157,108,141,158,108,166
2820 data247,232,169,0,202,240
2830 data43,24,105,1,201,10
2840 data240,3,76,72,105,169
2850 data1,24,109,157,108,201
2860 data58,240,6,141,157,108
2870 data76,70,105,169,48,141
2880 data157,108,169,1,24,109
2890 data156,108,141,156,108,76
2900 data70,105,24,105,48,141
2910 data158,108,96,165,113,141
2920 data240,2,165,114,141,241
2930 data2,32,232,104,32,232
2940 data104,96,173,242,2,32
2950 data232,104,173,243,2,32
2960 data232,104,24,173,246,2
2970 data109,242,2,141,242,2
2980 data173,247,2,109,243,2
2990 data141,243,2,96,32,243
3000 data104,32,56,105,32,125
3010 data105,32,142,105,160,0
3020 data169,108,133,140,169,151
3030 data133,139,177,139,240,7
3040 data32,232,104,200,76,196
3050 data105,169,0,32,232,104
3060 data32,216,105,96,173,240
3070 data2,133,141,173,241,2
3080 data133,142,162,0,165,113
3090 data129,141,230,141,165,114
3100 data129,141,96,169,0,141
3110 data245,2,141,244,2,141
3120 data237,2,169,108,133,140
3130 data169,59,133,139,32,137
3140 data103,165,248,32,38,106
3150 data32,210,255,165,248,32
3160 data54,106,32,210,255,165
3170 data247,32,38,106,32,210
3180 data255,165,247,32,54,106
3190 data32,210,255,96,74,74
3200 data74,74,24,105,48,201
3210 data58,176,1,96,24,105
```

```
3220 data7,96,10,10,10,10
3230 data32,38,106,96,32,125
3240 data105,32,142,105,169,153
3250 data32,232,104,169,34,32
3260 data232,104,96,32,194,106
3270 data32,62,106,165,254,10
3280 data144,15,169,18,32,232
3290 data104,169,34,32,232,104
3300 data169,34,32,232,104,198
3310 data113,208,2,198,114,160
3320 data1,169,108,133,140,169
3330 data133,133,139,177,139,240
3340 data10,32,232,104,200,238
3350 data244,2,76,121,106,169
3360 data1,141,237,2,76,24
3370 data103,173,237,2,208,5
3380 data169,34,32,232,104,169
3390 data0,32,232,104,32,216
3400 data105,169,0,32,232,104
3410 data169,0,32,232,104,96
3420 data173,237,2,208,5,169
3430 data34,32,232,104,169,0
3440 data141,236,2,32,232,104
3450 data32,216,105,96,169,34
3460 data32,232,104,169,59,32
3470 data232,104,169,0,32,232
3480 data104,32,216,105,96,147
3490 data32,32,142,32,32,32
3500 data32,151,18,32,32,76
3510 data46,77,46,32,32,42
3520 data42,32,80,82,73,78
3530 data84,32,76,79,65,68
3540 data69,82,32,42,42,32
3550 data32,13,13,32,32,32
3560 data32,32,32,32,32,18
3570 data40,67,41,39,56,56
3580 data32,67,79,77,77,46
3590 data80,82,79,70,69,83
3600 data83,73,79,78,65,76
3610 data13,13,13,31,73,78
3620 data68,73,82,73,90,90
3630 data73,32,73,78,32,32
3640 data32,32,32,32,32,32
3650 data32,32,32,32,13,72
3660 data69,83,65,68,69,67
3670 data32,79,32,68,69,67
3680 data73,77,65,76,69,32
3690 data40,72,47,68,41,63
3700 data0,13,13,76,79,67
3710 data65,90,73,79,78,69
3720 data32,73,78,73,90,73
3730 data65,76,69,32,63,32
3740 data32,0,13,13,76,79
3750 data67,65,90,73,79,78
3760 data69,32,70,73,78,65
3770 data76,69,32,32,63,32
3780 data0,13,13,78,85,77
3790 data69,82,79,32,68,69
3800 data76,76,65,32,80,82
3810 data73,77,65,32,82,73
3820 data71,65,32,66,65,83
3830 data73,67,32,32,32,32
3840 data32,32,32,32,13,40
3850 data18,82,69,84,85,82
3860 data78,146,32,80,69,82
3870 data32,49,48,48,48,48
3880 data41,63,32,0,13,13
3890 data73,78,67,82,69,77
3900 data69,78,84,79,32,76
3910 data73,78,69,69,32,66
3920 data65,83,73,67,13,40
3930 data18,82,69,84,85,82
3940 data78,146,32,80,69,82
3950 data32,49,48,41,63,32
3960 data0,147,17,17,17,17
3970 data17,17,17,17,17,29
3980 data29,29,29,29,29,29
3990 data29,67,82,69,65,90
4000 data73,79,78,69,32,32
4010 data32,76,73,78,69,69
4020 data32,32,66,65,83,73
4030 data67,13,13,29,29,29
4040 data29,29,29,29,29,29
4050 data68,65,76,76,39,73
4060 data78,68,73,82,73,90
4070 data90,79,32,32,32,32
4080 data32,32,0,19,17,17
4090 data17,17,17,17,17,17
```



```
4100 data17,17,17,29,29,29
4110 data29,29,29,29,29,29
4120 data29,29,29,29,29,29
4130 data29,29,29,29,29,29
4140 data29,29,32,0,4,68
4150 data72,13,20,12,48,49
4160 data50,51,52,53,54,55
4170 data56,57,13,20,18,48
4180 data49,50,51,52,53,54
4190 data55,56,57,65,66,67
4200 data68,69,70,13,20,34
4210 data199,40,51,52,41,199
4220 data40,51,52,41,199,40
4230 data50,48,41,34,0,151
4240 data50,48,57,44,48,54
4250 data48,58,151,50,49,48
4260 data44,48,48,51,58,151
4270 data50,49,49,44,48,58
4280 data151,50,49,51,44,50
4290 data53,53,0,
```

DATA... DATA...

Proseguiamo nell'opera di dare uniformità ai listati presentati da Commodore Professional

Su di un numero precedente della nostra rivista è stato pubblicato il programma FAST.LISTER con il quale si è definitivamente adottato uno standard per quanto riguarda la presentazione dei listati per i 'piccoli' della COMMODORE, tramite la codifica dei codici di controllo e degli altri caratteri particolari.
Volendo continuare l'opera di normalizzazione intrapresa presenteremo su questo numero una serie di programmi sia per C128 che per C64, che ci aiuteranno in futuro offrendoci la possibilità di proporVi listati più chiari e di una sempre più semplice copiatura.

Molti tra i nostri lettori, hanno contestato il modo con cui sono stati presentati alcuni trà i programmi da noi pubblicati. Effettivamente è difficile non rabbrividire davanti alla prospettiva di dover digitare un lungo listato contenente qualche migliaio di numeri in linee DATA, con un solo checksum di controllo alla fine, oppure addirittura mancante anche di quell'unica verifica.

Questa volta perciò ci occuperemo esclusivamente del codice che è possibile trasferire in linee DATA nei programmi basic, (qualsiasi natura esso abbia: potrebbero essere cioè programmi o routine in LM, oppure i dati di una serie di sprite, o ancora una schermata grafica, ecc.). Questi dati, possono essere contenuti come si diceva in linee data e quindi registrati assieme al programma basic che li gestisce, oppure, sarà possibile che costituiscano su disco un proprio file di tipo programma (formato BINARIO), che potrà essere caricato in memoria con le istruzioni LOAD, BLOAD BOOT ecc. In quest'ultimo caso, evitando il ricorso alle linee DATA, si risparmia sia tempo per il caricamento in memoria del codice, che alle volte, tempo per la digitazione (e successiva correzione) della parte di programma basic interessata, e questo si ottiene ricorrendo al classico formato MLX.

Per chi non sapesse di cosa si tratta, brevemente diremo che tramite questo formato, con un programma apposito, (MLX), è possibile immettere in memoria ed in seguito salvare su disco il codice che ci interessa, ottenendo inoltre il controllo sistematico per ogni linea di dati immessa con un dato di checksum che effettua il controllo della regolarità della linea. L'enorme vantaggio di cui subito ci si rende conto è che la fase (fastidiosissima) di ricontrollo dei dati alla fine della battitura, nel momento in cui ci si accorge di qualche errore, viene automaticamente eliminata!

La stessa facilitazione abbiamo voluto offrire per i casi in cui non risultasse comunque conveniente il ricorso al formato MLX, quando cioè i dati vengono pubblicati sotto forma di linee basic, (READ/DATA). Questo è stato ottenuto con la stesura di un ulteriore programma, che non è altro che un sofisticato editor finalizzato all'inserimento di linee basic contenenti esclusivamente dati numerici, tramite il quale, (disponendo di una serie di dati di checksum per ogni piccolo gruppo di linee data), si ricalca l'operato dell'MLX, effettuando questa volta il controllo della correttezza sul gruppo di linee e dando il consenso alla prosecuzione soltanto quando tutto è in regola.

Ma esaminiamo con calma uno per uno i listati ed impariamo ad usare le utility relative, soffermandoci in modo particolare sul breve ma flessibile MLX.DATAMAKER, in quanto una volta capito il concetto della presentazione dei dati (in un formato o nell'altro), sarà pressoché automatica la comprensione delle restanti utilities.

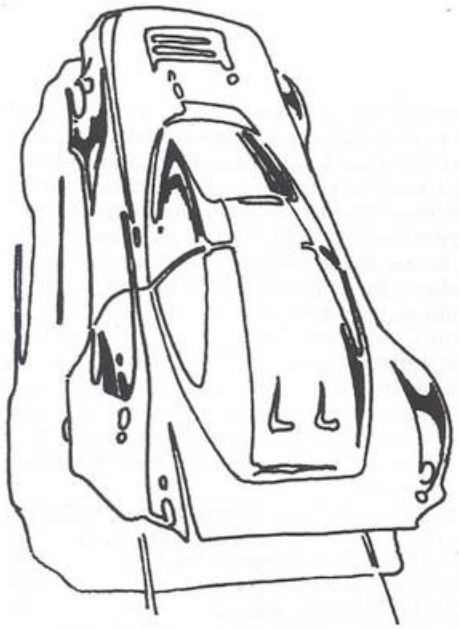
128mlx.datamaker

Come si evince dal nome, questo programma è un flessibilissimo datamaker, che offre la possibilità di generare linee data (in esa o dec), op-

pure creare listati nel formato MLX, compatibile con il formato del codice MLX pubblicato da SUPERCOMMODORE 64&128.

Per quanto riguarda il formato MLX, ogni linea inizia presentando come numero di linea la locazione di memoria alla quale si riferisce il primo data in essa contenuto, prosegue con sei data in decimale e termina con un dato di controllo che tiene conto anche del numero di linea.

Il DATAMAKER che consente invece di generare linee data delle quali è possibile variare i normali parametri, (numerazione, step, numero di data per linea, formato decimale o esadecimale), offre anche la possibilità di calcolare il checksum totale alla fine dei dati, oppure dei codici di controllo intermedi, per blocchi di linee.



Il breve listato, si divide in tre sezioni:

01/13 Inizializzazione ed input dei parametri.

15/28 Costruzione linee data formato checksum.

30/39 Costruzione linee data formato MLX.

I parametri richiesti sono i seguenti:

— Formato MLX: è accettata in risposta solo una s (SI) o una n (NO). In caso di risposta affermativa viene posta a uno la variabile MX (flag).

— Locazione iniziale: viene immesso nella variabile M1 il valore della locazione di memoria da cui avrà inizio il lavoro delle due sezioni seguenti. Lo stesso valore, viene convertito nella stringa M1$, predisponendo un'input successivo. Si controlla inoltre che il valore digitato non sia inferiore a $0400, in quanto non avrebbe senso l'elaborazione di questa parte della memoria del 128, sia per il banco zero che per il banco 1.

— Locazione finale: questo parametro viene immagazzinato nella variabile M2, non prima di aver controllato che non sia uguale od inferiore alla variabile M1 e che non sia superiore a $FEFF, locazione oltre la quale si hanno sia nel banco zero che nel banco uno i registri di configurazione della memoria (MMU).

— Numero banco: alla variabile NB si assegnerà il numero del banco (0-1), da cui prelevare il codice.

A questo punto, (linea 7), se la variabile MX esiste, (se cioè ha un qualche valore diverso da zero), significa che si è optato per il formato MLX, per cui l'input dei parametri è terminato e si salta alla linea 30. In caso contrario si provvede al prelievo di ulteriori parametri:

— Dec o esa: Per prima cosa si dovrà scegliere se i data dovranno essere decimali o meno. Per difetto si assumerà che i dati saranno esadecimali, variabile B$ uguale al carattere "e". Verrà poi definita la variabile DM (massimo numero di data per una fisica di schermo).

— Linea di inizio: Nella variabile LN verrà posto il numero di linea basic iniziale delle linee create dal datamaker. Viene assunta per default come numero di linea iniziale il valore di M1 (locazione di inizio), precedentemente convertito nella stringa M1$. Si controlla inoltre che il numero digitato non sia inferiore a 50 per evitare la sovrapposizione delle linee create con le linee basic del datamaker.

— Data per linea: Si assume per difetto che il numero di data per ogni linea, sia tale da non superare le quaranta colonne: (7 dati in base decimale oppure 10 dati esa.). La variabile DL conterrà il valore digitato.

— Incremento: Lo step di linea, verrà proposto uguale al numero di data per linea, in modo tale che se la variabile LN, (numero di linea iniziale) è uguale a M1 (locazione di memoria iniziale), come suggerito dal datamaker, ogni numero di linea rappresenterà la locazione di memoria del primo dato in essa contenuto. Dopo questo input, viene inoltre controllato che in caso di immissione di valori incongruenti, il programma non cada in errore durante l'elaborazione, per un numero di linea troppo grande. Il basic infatti accetta come numero di linea più elevato 63999.

— Checksum totale o a blocchi: L'ultima opzione riguarda la scelta del checksum. Ossia, se questo debba essere unico per tutti i data, (in caso di brevi routine), oppure se si voglia creare dei dati di controllo intermedi per piccoli gruppi di linee.

— Numero di linee per blocco: Questa domanda sarà posta solo se si sarà optato per il checksum a blocchi, e la variabile LB conterrà il numero di linee per ogni blocco.

Inizierà a questo punto il lavoro vero e proprio del nostro DATAMAKER, che è simile nel funzionamento a quello della sezione MLX, per cui essendo entrambe estremamente semplici, eviteremo di dilungarci, soffermandoci soltanto su alcune osservazioni.

Alla linea 15 e 30 vengono analogamente inizializzate alcune variabili. La linea 16 provvede a stampare il numero di linea dopo aver incrementato il contatore L1. Osserviamo la particolarità che se L1 ha raggiunto il valore LB, quella linea dovrà contenere il CHECKSUM, per cui il numero di linea sarà incrementato di una sola unità rispetto all'ultimo numero di linea, lasciando così inalterato lo step tra le linee contenenti i data. Anche con i checksum intermedi cioè, i numeri di linea assumeranno sempre il valore della locazione di memoria del primo data in esse contenuto.

Nella 17 osserviamo che il dato di checksum sarà preceduto sulla linea da un asterisco, (*). Questo tornerà utile alla routine di lettura dei dati, come avremo modo di vedere.

Alla linea 23 possiamo invece notare che se siamo giunti alla fine del lavoro, (M1>M2), il dato di controllo, sarà questa volta preceduto da due asterischi: (**).

Per entrambe le sezioni, si fa ampio uso della tecnica del buffer della tastiera per appendere le linee create in coda al MLX.DATAMAKER. Le linee vengono infatti formate sullo schermo e sotto di esse viene scritta una istruzione GOTO, che rimanderà al punto opportuno del programma, dopo che alla linea 39, si sarà provveduto ad immettere nel buffer di tastiera un CHR$(19): HOME, e tre CHR$(13): RETURN. L'istruzione END sulla stessa linea provvede a terminare il programma, che dopo l'immissione diretta in memoria della linea basic, grazie al GOTO... riprende l'elaborazione. Questo ciclo si ripete fino a che giunti alla fine, invece del goto si ritroverà l'istruzione DELETE-...., che provvederà a cancellare dalla memoria il programma generatore. Sarà così possibile salvare le linee data create su disco, in modo da poterle fondere ad altri programmi tramite una delle tante utility adatte allo scopo.

Un'ultima osservazione per quanto riguarda il calcolo del CHECKSUM del MLX. Alla linea 31, il 'numero di lineà (valore della locazione di memoria) viene ridotto ad un numero compreso da 0 a 255, e salvato nella variabile V. Per ogni linea, il vettore V(5) conterrà il valore della relativa cella di memoria, ed

alla linea 35, si calcola il CK, (tenendo conto anche del numero di linea trasformato nella variabile V), facendolo sempre rientrare nel range 0-255.

C64mlx.datamaker

La versione per 64 di questo programma, ricalca quanto più fedelmente possibile quella per 128, anche se non si è potuto mantenere la stessa numerazione, per cui si rimanda a quanto detto più sopra, con alcune differenze. Nell'input dei parametri all'inizio, non sarà naturalmente posta la domanda sul numero di banco, e per quanto riguarda la memoria elaborabile, vengono accettati valori entro il range 251-65535, in pratica tutta quella disponibile, ad eccezione della pagina zero. La trasformazione da decimale in esadecimale, non essendo disponibile la funzione hex() del 128, si ottiene tramite la definizione di una propria funzione alla linea 3: ESA(x).

La differenza più importante è però dettata dal diverso modo di operare delle due macchine:

Si è detto infatti che il principio di funzionamento del datamaker, sta nel fatto che le linee data create si appendono in coda al programma tramite la tecnica del buffer della tastiera. Come tutti sanno però, mentre con il 128 grazie alla gestione della memoria a banchi non sorge alcun problema, con il 64, ad ogni immissione di una nuova linea, il sistema operativo, dovendo risistemare i puntatori di fine programma, effettua un CLR grazie al quale si perde il valore di tutte le le variabili definite. Il problema viene risolto scrivendo sullo schermo oltre alla linea data creata, anche una linea data con il valore attuale delle variabili che si devono mantenere, (nel nostro caso si tratta della linea con numero zero), dalla quale, con una chiamata alla subroutine in linea 13, ad immissione avvenuta, prima di reiniziare il ciclo, verranno ripristinati i precedenti valori.

Alla fine non disponendo dell'istruzione DELETE per cancellare il programma generatore, se ne cerca la fine, individuata dal numero 65 (ultima linea del datamaker, ulteriore motivo per non cambiare la numerazione del programma), e si modifica il puntatore all'inizio del testo basic (locazioni 43 e 44). Le linee dalla 61 alla 64, assolvono a quanto appena esposto, per cui non rimarrà altro che effettuare un SAVE e resettare i puntatori tramite lo spegnimento e riaccensione del computer, oppure con l'istruzione seguente:

POKE43,1:POKE44,8:NEW <RETURN>

128mlx.timestore

È questo il programma MLX vero e proprio che serve ad immettere da tastiera il codice listato, creato nel formato MLX con il DATAMAKER. I tasti operanti durante l'immissione dei dati, (che dovranno essere sempre premuti assieme al tasto SHIFT), sono: "C" = Catalog, "D" = Display: mostra la memoria sullo schermo, "N" = Nuovo indirizzo: serve a modificare l'indirizzo da cui immettere i dati, "L" = Load file, "S" = Save file. Il file viene salvato come file di programma in codice binario e data la oramai diffusissima presenza del drive a fianco di ogni c128, non si è volutamente previsto l'uso del registratore a cassette.

Una particolarità abbastanza rilevante, di questo programma, è che è possibile avere come area di lavoro, tutta la ram disponibile, del c128 e cioè: per il banco zero la memoria che va da $0b00 a $0f00 e da $1300 a $feff, e per il banco 1 la memoria che va da $0400 a $feff. È infatti il programma stesso che intelligentemente provvederà ad auto-rilocarsi, una volta terminata la iniziale fase di input dei parametri, se la memoria di lavoro coincide con la memoria occupata dal codice basic (per il banco zero), o a modificare in modo opportuno i puntatori per le variabili basic se l'area di lavoro nel banco 1, dovesse entrare in conflitto con lo spazio occupato dalle variabili del programma stesso.

A questo scopo, sarà opportuno non togliere dal drive il disco contenente il programma 128MLX.TIMESTORE, fino a che non si sarà superata questa fase, in quanto se sarà necessario, il rilocamento del programma verrà effettuato tramite la modifica dei relativi puntatori ed il ricaricamento dal disco in memoria del programma stesso.

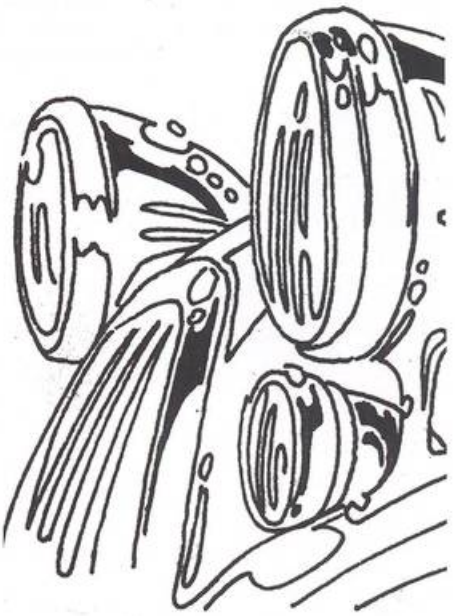
Altra particolarità degna di nota è il fatto che tra i parametri richiesti allo start verrà richiesto tra l'altro, un tempo (da uno a trenta minuti, ed ecco spiegato il perché del suffisso "TIME" al nome 128MLX), per lo step del save, e durante il lavoro, allo scadere del tempo viene effettuato un salvataggio, con un numero progressivo in testa al nome del file (da 00 a 99), per salvaguardare il lavoro in caso di interruzioni alla alimentazione. Al termine della copiatura sarà opportuno provvedere alla eliminazione dal disco di lavoro di tutti i file meno che quello con il numero progressivo più elevato.

Uno sguardo al listato

Una breve analisi del listato relativo evidenzia nelle prime tre linee di programma dei codici data che servono ad una breve routine per l'accensione e lo spegnimento del cursore sia sulle ottanta che sulle quaranta colonne. Sarà necessario ricopiare tale e quale anche la prima linea con la rem seguita dalle virgolette e da un certo numero di asterischi, in quanto sarà proprio questo lo spazio occupato dal codice suddetto.

Dopo una inizializzazione, (linee 100-134), si passa all'input dei parametri i quali sono: il numero del banco dove immagazzinare il codice, il nome del file col quale salvare il codice, lo step dei minuti per l'operazione di salvataggio, il numero iniziale del file (0-99), utile nel caso di copiatura in più sedute, l'indirizzo iniziale e l'indirizzo conclusivo che assieme al numero del banco, saranno pubblicati di volta in volta assieme al codice relativo.

Terminato l'input dei parametri, si



controlla (linee 206-240) che l'area tra l'indirizzo iniziale e l'indirizzo conclusivo non abbia ad entrare in conflitto con la memoria del codice basic o delle variabili di 128MLX ed in caso si prendono i dovuti provvedimenti. Nel caso fosse necessaria una rilocazione del programma basic, (linee 228-236) i parametri vengono temporaneamente salvati ed un flag (locazione 254) viene settato a 128 (linee 452-460), vengono scritte sullo schermo le relative istruzioni POKE e l'istruzione per il reboot del programma, che alla successiva inizializzazione, grazie al flag sopracitato, verrà saltata la fase di input dei parametri, i quali saranno recuperati tramite le linee 122-132. È opportuno notare che alla linea 134, si ridefinisce con stringhe vuote (" ") i tasti funzione da uno a otto, mentre per il nono ed il decimo

tasto funzione (HELP e RUN/STOP+SHIFT), lo stesso scopo viene raggiunto dalle poke 4104,0 e poke 4105,0 con le quali si pone a zero la lunghezza delle stringhe assegnate ai tasti relativi.

Alla linea 252 ha inizio il loop del programma principale, e dopo aver scritto sullo schermo la locazione di memoria relativa al primo dato della linea, si chiama per sette volte, (sei dati più il checksum), la routine di input di un dato (GOSUB 282) che verrà immagazzinato nel vettore a(7). Se al ritorno, il valore della variabile A (codice ASCII ultimo tasto premuto) è maggiore di 128 (carattere+SHIFT), ci troviamo di fronte alla richiesta di un comando, che viene richiamato alla linea 258.

Nella linea 266 si calcola il checksum della linea e si controlla che ci sia la dovuta corrispondenza con il codice di controllo immesso; in caso positivo si provvede ad immettere in memoria i sei dati ricevuti in input, si controlla il tempo trascorso dopo l'ultimo azzeramento della variabile TI$ e se è il caso si effettua il save. Il salvataggio automatico allo scadere del tempo prestabilito può essere inibito di volta in volta per ogni linea premendo il tasto ESCAPE all'opportuno segnale, oppure può temporaneamente essere disabilitato mettendo in posizione ON il tasto CAPS/LOCK.

Dalla linea 282 alla linea 328, abbiamo la subroutine per l'input di un dato la quale si occupa anche della gestione della visualizzazione dei caratteri sullo schermo. Da notare che oltre al controllo tramite il checksum alla fine della linea, per ogni carattere digitato c'è un'ulteriore controllo, in modo che non sarà mai possibile digitare un numero più grande di 255. È noto infatti che l'errore più frequente che si fa durante la battitura è lo scambio di due caratteri adiacenti. Con questo controllo, una parte di questi errori vengono eliminati: non sarà possibile infatti digitare 321 al posto di 231, oppure immettere 271 al posto di 217, in quanto, al momento che si dovesse premere una cifra più alta di due come primo carattere, oppure una cifra più alta di 5 nel secondo caso verrebbe segnalato un'errore. Le linee da 312 a 320 provvedono a questi controlli.

Come già detto, il file programma creato con 128MLX.TIMESTORE potrà esere caricato in memoria con le apposite istruzioni LOAD, BLOAD oppure BOOT.

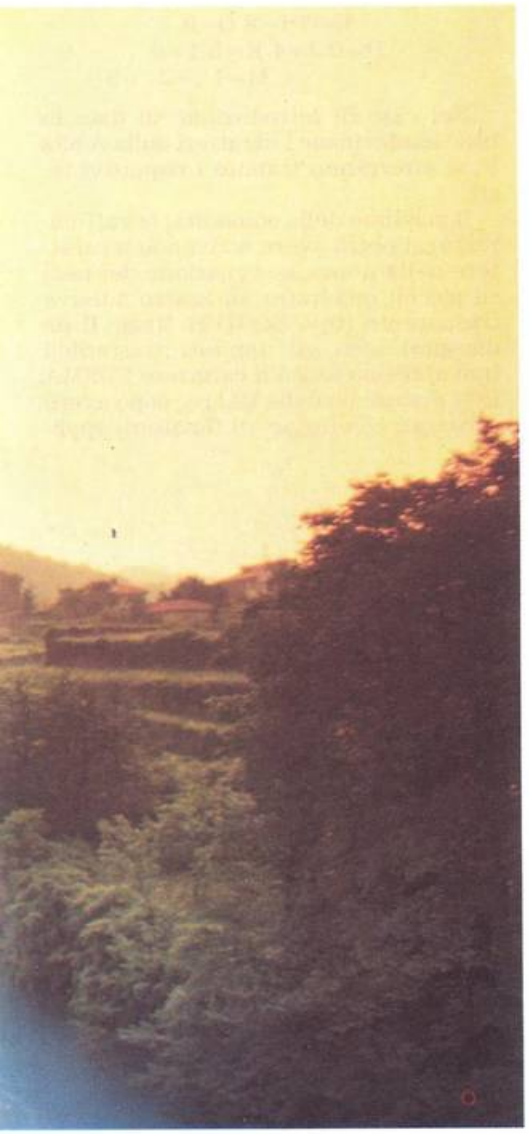
128.dataenter

Dicevamo poco sopra, che con il programma MLX.DATAMAKER, oltre a linee in formato MLX è possibile creare listati di data ben ordinati, con i loro cheksum ogni tante linee, eccetera.

Però, mentre nel caso di immissione di codice in formato MLX, tramite il programma MLX.TIMESTORE è possibile un controllo diretto durante la digitazione, che non richiede alcuna fati-

ca, la copiatura di linee data per ordinate che siano è causa comunque di un notevole sforzo, anche se il basic 7.0 del c128 mette a disposizione la comoda AUTO tramite la quale si evita di scrivere il numero di linea, ed i tasti funzione, facilmente ridefinibili per la parola chiave DATA ed anche per la virgola. Nonostante ciò, specialmente per i data in esadecimale, non si sa se sia più convieniente usare il tastierino o perdersi sulla tastiera, sanza contare il lavoro di ricerca e correzione degli errori alla fine della copiatura!

In questo caso ci sarà senz'altro di aiuto il programma DATAENTER, che non è altro che un sofisticato editor di linee data, parente stretto dell'MLX, in quanto non fa altro che darci una mano quando si tratta di immettere lunghe sequenze di linee data, esonerandoci dalla digitazione del numero di linea, della parola chiave DATA, delle virgole di separazione tra i dati, ed infine, ogni volta che si arriva ad un cheksum, il programma effettua il controllo della parte digitata, e se tutto è in ordine il gruppo di linee data viene immesso in memoria e contemporaneamente si provvede a cambiarne il colore sullo schermo, dopodiché è possibile procedere.



In caso contrario, viene richiesta la correzione, riportndo il cursore sul primo dato della prima linea del gruppo. Per evidenziare il fatto che c'è bisogno di una correzione, il gruppo di linee corrente cambierà ancora una volta colore. Si possono avere perciò tre condizioni di lavoro, a cui corrispondono altrettanti colori del testo sullo schermo:

Immissione del testo:	VERDE
Correzione del testo:	GIALLO
Testo immesso in memoria:	BLEU

Nel caso di errore quindi, sarà possibile muoversi con i tasti cursore lungo le cifre dei data della linea, per posizionarsi sulla cifra errata ed apportare la correzione. Premendo RETURN si passa alla verifica e correzione della linea successiva, e così di seguito fino alla linea contenente il cheksum. Premendo RETURN in corrispondenza di quest'ultimo, dopo averlo eventualmente corretto, se l'errore è stato eliminato, sarà possibile proseguire, altrimenti il ciclo si ripeterà fino a che il dato di controllo digitato non corrisponderà alla somma dei dati del gruppo corrente, solamente allora si avrà il nulla osta per proseguire con un'altro gruppo di linee.

Anche in questo caso, si userà la tecnica di appendere le linee basic DATA, in coda al programma 128.DATAENTER, scrivendo la linea stessa sullo schermo ed effettuando le opportune POKE nel buffer della tastiera. Analogamente al DATAMAKER, il solito GOTO.... farà riprendere il controllo al programma nel punto opportuno.

Questo però solo per quanto riguarda 128.DATAENTER, in quanto per la versione relativa al C64, a causa dei motivi menzionati in precedenza in riferimento al datamaker, (perdita delle variabili alla immissione di una nuova linea basic), si èdovuto adottare un'ulteriore stratagemma. Nel presente programma infatti sono tante le variabili da dover 'ricordare', per cui non è applicabile la tecnica adottata in precedenza. Il problema, si risolve scrivendo ogni linea o gruppo di linee in un file sequenziale su disco, e solo dopo la completa copiatura del listato, una sezione del programma, di pochissime righe, e funzionante senza il bisogno di tenere in memoria alcuna variabile (o meglio, si dovrà salvare soltanto la variabile di stato ST, per riconoscere la fine del file) si occuperà di leggere il file ed appendere in memoria le linee data, con la solita tecnica.

Come al solito, dopo una breve inizializzazione, verranno richiesti alcuni parametri per il corretto funzionamento del programma:

Numero di linea di inizio. Verrà accettato qualsiasi valore superiore a 1000, per non entrare in conflitto con la numerazione del programma di gestione.

Verrà a questo punto richiesto se i dati in ingresso saranno in decimale od esa, e quanti data per ogni linea. Per omogeneità al datamaker, si definisce per default che il numero dei dati su ogni linea sia tale da non far superare la lunghezza di quaranta caratteri alla linea stessa, mentre non sarà accettato un numero di dati tale che renderebbe la linea basic più lunga dei 160 caratteri permessi per il 128, oppure 80 nel caso del 64.

La quarta informazione che si dovrà fornire al programma sarà l'incremento per i numeri di linea (non sarà accettato un valore inferiore a 2).

Si dovrà quindi indicare se il listato presenta un'unico checksum alla fine oppure se il dato di controllo è ricorrente ogni tante linee. Solo in quest'ultimo caso, si procederà alla richiesta del numero di linee per gruppo.

Note per l'uso di 128.DATAENTER

Il programma, (come anche gli altri per il 128), funziona sia sulle quaranta che sulle ottanta colonne, usando in quest'ultimo caso il modo fast. È estremamente automatico, e non resta da fare altro che immettere i dati uno dopo l'altro, senza curarsi nè dei numeri di linea, nè della parola riservata DATA, nè delle virgole e tantomeno del passaggio alla riga seguente, in quanto una volta immesso il numero di dati determinato si passerà automaticamente alla linea successiva. Durante l'immissione, sarà possibile usare il tasto DELETE per cancellare alcuni caratteri già immessi, ed effettuare correzioni.

Terminato di editare un gruppo di linee, verrà presentata la linea che dovrà contenere il dato di controllo, riconoscibile perché subito dopo la parola DATA, avrà un'asterisco seguito dalla virgola. Per confermare la completa immissione del checksum bisognerà premere RETURN.

Se tutto è in ordine si proseguirà, altrimenti come già spiegato si dovrà procedere alla correzione. In questa fase, sarà possibile spostarsi con i tasti cursore RIGHT e LEFT per posizionare il cursore sull'errore. Premere RETURN per passare alla linea successiva.

Alla fine del lavoro, prima di immettere l'ultimo dato di controllo, premere <*>, il che avrà lo scopo di far capire al programma che il lavoro è terminato, e

```
end of pass 1, 0 errors, * = $c169
                00100 ; scansione tastiera e ridefinizione
                00102 ; caratteri per tastierino numerico.
                00104 ;
                00106          *=$c000          ;origine a 49152
                00108 .obj*
                00110 .out     "scan.tast"      ;output del codice su disco
c000- a9 00     00112 start    lda #$00
c002- 85 cc     00114          sta $cc          ;accende il cursore
c004- 85 c6     00116          sta $c6          ;e azzera buffer tastiera
c006- a5 c6     00118 tasto    lda $c6          ;e' stato premuto un tasto?
c008- f0 fc     00120          beq tasto        ;se no aspetta.
c00a- ad 77 02  00122          lda $0277        ;preleva il carattere
c00d- a6 fc     00124          ldx $fc          ;flag: 0=dec 1=esa
c00f- f0 0b     00126          beq a0           ;input decimale
                00128 ;
c011- c9 41     00130          cmp #"a          ;tasti a/f per input esa.
c013- 30 07     00132          bmi a0
c015- c9 47     00134          cmp #"g
c017- 10 03     00136          bpl a0
c019- 4c 74 c0  00138          jmp r1
                00140 ;
                00142 ;simula tastierino numerico con i tasti hm,.jkluio
                00144 ;
c01c- c9 48     00146 a0       cmp #"h          ;tasto h=0
c01e- d0 04     00148          bne a1
c020- a9 30     00150          lda #"0
c022- d0 50     00152          bne r1
c024- c9 4d     00154 a1       cmp #"m          ;tasto m=1
c026- d0 04     00156          bne a2
c028- a9 31     00158          lda #"1
c02a- d0 48     00160          bne r1
c02c- c9 2c     00162 a2       cmp #",          ;tasto ,=2
c02e- d0 04     00164          bne a3
c030- a9 32     00166          lda #"2
c032- d0 40     00168          bne r1
c034- c9 2e     00170 a3       cmp #".          ;tasto .=3
c036- d0 04     00172          bne a4
c038- a9 33     00174          lda #"3
c03a- d0 38     00176          bne r1
c03c- c9 4a     00178 a4       cmp #"j          ;tasto j=4
c03e- d0 04     00180          bne a5
c040- a9 34     00182          lda #"4
c042- d0 30     00184          bne r1
c044- c9 4b     00186 a5       cmp #"k          ;tasto k=5
c046- d0 04     00188          bne a6
c048- a9 35     00190          lda #"5
c04a- d0 28     00192          bne r1
c04c- c9 4c     00194 a6       cmp #"l          ;tasto l=6
c04e- d0 04     00196          bne a7
c050- a9 36     00198          lda #"6
c052- d0 20     00200          bne r1
c054- c9 55     00202 a7       cmp #"u          ;tasto u=7
c056- d0 04     00204          bne a8
c058- a9 37     00206          lda #"7
c05a- d0 18     00208          bne r1
c05c- c9 49     00210 a8       cmp #"i          ;tasto i=8
c05e- d0 04     00212          bne a9
c060- a9 38     00214          lda #"8
c062- d0 10     00216          bne r1
c064- c9 4f     00218 a9       cmp #"o          ;tasto o=9
c066- d0 04     00220          bne cifre
c068- a9 39     00222          lda #"9
c06a- d0 08     00224          bne r1
                00226 ;
c06c- c9 30     00228 cifre    cmp #"0          ;tasti 0/9 input esa e dec.
c06e- 30 15     00230          bmi data
c070- c9 3a     00232          cmp #":
c072- 10 11     00234          bpl data
                00236 ;
                00238 ;carica in x il flag per salto 'on goto' dal basic e torn
                00240 ;
c074- a2 01     00242 r1       ldx #1           ;carattere dec o esa
c076- d0 0a     00244          bne rrit
c078- a2 02     00246 r2       ldx #2           ;asterisco, return
c07a- d0 06     00248          bne rrit
c07c- a2 03     00250 r3       ldx #3           ;delete, left
c07e- d0 02     00252          bne rrit
c080- a2 04     00254 r4       ldx #4           ;right, asterisco o stop
c082- 4c c5 c0  00256 rrit     jmp ritorno
                00258 ;
                00260 ;controlla tasti per funzioni particolari
                00262 ;nelle tre situazioni dell'input
                00264 ;
c085- a6 fb     00266 data     ldx $fb          ;controlla il flag di input
c087- d0 0f     00268          bne cksum        ;e se e' zero continua
c089- c9 2a     00270          cmp #"*          ;asterisco?
c08b- f0 eb     00272          beq r2
c08d- c9 14     00274          cmp #20          ;delete?
c08f- f0 eb     00276          beq r3
c091- c9 ae     00278          cmp #"¬          ;cmdr+s? (stop)
c093- f0 eb     00280          beq r4
c095- 4c 00 c0  00282          jmp start        ;cicla
                00284 ;
c098- a6 fb     00286 cksum    ldx $fb          ;controlla flag cksum
c09a- e0 01     00288          cpx #1           ;se non e' uguale a uno
c09c- d0 0f     00290          bne correz.      ;salta
c09e- c9 2a     00292          cmp #"*          ;asterisco?
c0a0- f0 de     00294          beq r4
```

di comportarsi di conseguenza. L'ultimo checksum sarà infatti preceduto da due asterischi invece che da un solo. Questa operazione, sarà da fare sia che l'ultimo checksum coincida con il checksum di gruppo, sia pure nel caso che l'ultimo gruppo contenga un numero minore di dati. Digitando un asterisco durante l'immissione dei dati infatti si passerà automaticamente all'input dell'ultimo checksum di controllo.

Chi possiede il 128, potrà usufruire del tastierino, sia per per l'immissione dei dati in base decimale che per quelli in esadecimale, senza dover rincorrere le lettere sulla tastiera, in effetti vengono ridefiniti i tasti "F1"=A, "F3"=B, "F5"=C, "F7"=D, "+"(più)=E, "-"(meno)=F.

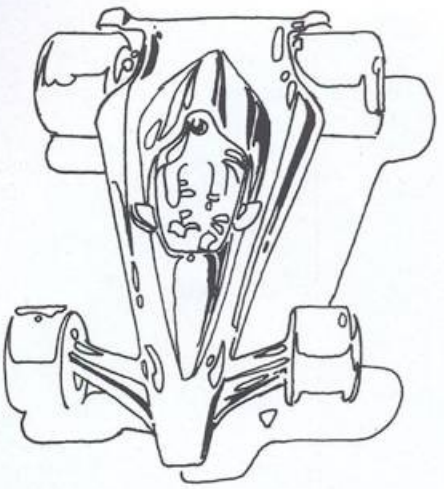
Anche i possessori del C64 avranno la possibilità di fruire del vantaggio dell'uso del tastierino numerico, si potrà infatti usare sia i normali tasti numerici che le lettere alfabetiche ridefinite secondo la seguente tabella.

0

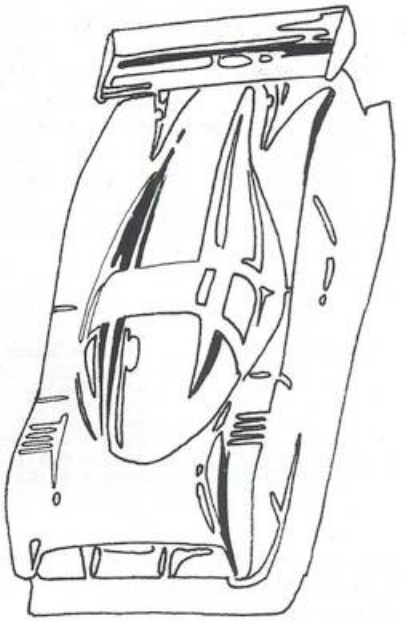
U=7 I=8 O=9
H=0 J=4 K=5 L=6
M=1 ,=2 .=3

Nel caso di introduzione di data in base esadecimale i caratteri dalla A alla F, si otterranno tramite i rispettivi tasti.

Il massimo della comodità, (e raffinatezza) si potrà avere scrivendo il carattere della nuova assegnazione dei tasti su piccoli quadratini di nastro adesivo trasparente (tipo SCOTCH Magic II da disegno), con gli appositi trasferibili (noi abbiamo usato il carattere FORMA RQ4 maiuscolo della R41) e, dopo averli spruzzati con un po' di fissatore, appli-



```
c0a2- c9 14     00296          cmp #20         ;delete?
c0a4- f0 d6     00298          beq r3
c0a6- c9 0d     00300          cmp #13         ;return?
c0a8- f0 ce     00302          beq r2
c0aa- 4c 00 c0  00304          jmp start       ;cicla
                00306 ;
c0ad- a6 fb     00308 correz.  ldx $fb         ;controlla se il flag
c0af- e0 02     00310          cpx #2          ;indica la fase di correzione
c0b1- f0 03     00312          beq corr.1
c0b3- 4c 00 c0  00314          jmp start       ;altrimenti cicla
c0b6- c9 0d     00316 corr.1   cmp #13         ;return?
c0b8- f0 be     00318          beq r2
c0ba- c9 1d     00320          cmp #29         ;crsr left?
c0bc- f0 be     00322          beq r3
c0be- c9 9d     00324          cmp #157        ;crsr right?
c0c0- f0 be     00326          beq r4
c0c2- 4c 00 c0  00328          jmp start
                00330 ;
                00332 ;il tasto premuto e' valido, per cui torna
                00334 ;
c0c5- 85 fd     00336 ritorno  sta $fd         ;salva il carattere
c0c7- 86 fe     00338          stx $fe         ;ed imposta flag per on goto
c0c9- a2 00     00340          ldx #0
c0cb- 8e 77 02  00342          stx $0277       ;azzera coda buffer
c0ce- 78        00344          sei             ;e dopo aver controllato
c0cf- a9 14     00346          lda #$14        ;che il cursore non sia
c0d1- 85 cd     00348          sta $cd         ;nella fase di acceso
c0d3- a4 d3     00350          ldy $d3         ;nel qual caso
c0d5- 46 cf     00352          lsr $cf         ;inverte la cella in reverse
c0d7- ae 87 02  00354          ldx $0287
c0da- b1 d1     00356          lda ($d1),y
c0dc- b0 11     00358          bcs ea5c
c0de- e6 cf     00360          inc $cf
c0e0- 85 ce     00362          sta $ce
c0e2- 20 24 ea  00364          jsr $ea24
c0e5- b1 f3     00366          lda ($f3),y
c0e7- 8d 87 02  00368          sta $0287
c0ea- ae 86 02  00370          ldx $0286
c0ed- a5 ce     00372          lda $ce
c0ef- 29 7f     00374 ea5c     and #%01111111
c0f1- 20 1c ea  00376          jsr $ea1c
c0f4- a9 01     00378          lda #$01        ;dopodiche'
c0f6- 85 cc     00380          sta $cc         ;spegne il cursore
c0f8- 58        00382          cli
c0f9- 60        00384          rts
                00386 ;
                00388 ; ************ end ***************
                00390 ;
                00392 ; ******** beep errore **********
                00394 ;
c0fa- a9 0f     00396 berr     lda #15
c0fc- a2 2d     00398          ldx #45
c0fe- a0 a5     00400          ldy #165
c100- 20 3f c1  00402          jsr bep1
c103- a9 21     00404          lda #33
c105- a2 06     00406          ldx #06
c107- a0 05     00408          ldy #05
c109- 20 49 c1  00410          jsr bep2
c10c- a2 ff     00412          ldx #255
c10e- 4c 53 c1  00414          jmp beep
                00416 ;
                00418 ; ******** beep ok! **************
                00420 ;
c111- a9 0f     00422 beok     lda #15
c113- a2 00     00424          ldx #00
c115- a0 f7     00426          ldy #247
c117- 20 3f c1  00428          jsr bep1
c11a- a9 11     00430          lda #17
c11c- a2 28     00432          ldx #40
c11e- a0 00     00434          ldy #00
c120- 20 49 c1  00436          jsr bep2
c123- a2 64     00438          ldx #100
c125- 4c 53 c1  00440          jmp beep
                00442 ;
                00444 ; ******** bip tasto *************
                00446 ;
c128- a9 0f     00448 bipt     lda #15
c12a- a2 32     00450          ldx #50
c12c- a0 f7     00452          ldy #247
c12e- 20 3f c1  00454          jsr bep1
c131- a9 11     00456          lda #17
c133- a2 c8     00458          ldx #200
c135- a0 00     00460          ldy #00
c137- 20 49 c1  00462          jsr bep2
c13a- a2 32     00464          ldx #50
c13c- 4c 53 c1  00466          jmp beep
                00468 ;
                00470 ;******** subroutine beep *********
                00472 ;
c13f- 8d 18 d4  00474 bep1     sta $d418
c142- 8e 05 d4  00476          stx $d405
c145- 8c 06 d4  00478          sty $d406
c148- 60        00480          rts
                00482 ;
c149- 8d 04 d4  00484 bep2     sta $d404
c14c- 8e 01 d4  00486          stx $d401
c14f- 8c 00 d4  00488          sty $d400
c152- 60        00490          rts
                00492 ;
```



carli sull'angolo in basso a destra dei rispettivi tasti. Per ottenere un buon esito, è consigliabile attaccare una striscia di scotch ad una superfice estremamente liscia e dura (una lastra di vetro), tagliare i quadratini con una lametta o taglierino, scriverci i caratteri voluti ed alla fine spruzzarli col fissatore. A questo punto sarà molto facile staccarli sempre con la lametta o con un coltellino e posarli sui tasti di destinazione, dove si fisseranno con una leggera pressione.

Le lettere aggiunte, a questo punto sembreranno serigrafate sui tasti come tutte le altre: dalla foto si può osservare la modifica apportata al tastierino di un 128.

Non ci crederete, ma l'uso del programma DATAENTER per la copiatura di linee data (specialmente se esadecimali), dopo un poca di pratica potrà dimezzare i tempi di lavoro, senza contare che alla fine si potrà essere matematicamente certi che il listato sarà senz'altro privo di errori di battitura!

Per l'immissione di listati in più sedute, si dovrà interrompere il programma (premendo il tasto COMMODORE+S=STOP) subito dopo l'immissione di un checksum, e nel caso di 128.DATAENTER, salvare il programma con tutte le linee DATA già immesse in coda. Alla ripresa del lavoro, dopo il ricaricamento in memoria, alla richiesta dei parametri, sarà sufficente indicare come linea di inizio il numero della linea successiva al checksum in cui eravamo arrivati, e riprendere il lavoro.

Al termine della copiatura, il programma per 128, proporrà il cursore sull'istruzione per la cancellazione del 128.DATAENTER, lasciando in memo-

```
c153- a0 ff     00494 beep    ldy #255       ;loop ritardo
c155- 88        00496 rit1    dey
c156- d0 fd     00498         bne rit1
c158- ca        00500         dex
c159- d0 f8     00502         bne beep
                00504 ;
c15b- a9 10     00506         lda #16        ;spegne il sid
c15d- 8d 04 d4  00508         sta $d404
c160- a9 00     00510         lda #0
c162- 8d 01 d4  00512         sta $d401
c165- 8d 00 d4  00514         sta $d400
c168- 60        00516         rts
end of pass 2, 0 errors, * = $c169


--- label file ---


start = $c000       tasto = $c006       a0 = $c01c          a1 = $c024
a2 = $c02c          a3 = $c034          a4 = $c03c          a5 = $c044
a6 = $c04c          a7 = $c054          a8 = $c05c          a9 = $c064
cifre = $c06c       r1 = $c074          r2 = $c078          r3 = $c07c
r4 = $c080          rrit = $c082        data = $c085        cksum = $c098
correz. = $c0ad     corr.1 = $c0b6      ritorno = $c0c5     ea5c = $c0ef
berr = $c0fa        beok = $c111        bipt = $c128        bep1 = $c13f
bep2 = $c149        beep = $c153        rit1 = $c155
```

ria soltanto le linee data aggiunte che potranno così essere salvate oppure fuse al programma di gestione con una delle tante utility di merge.

La versione per 64, avrà invece creato un file sequenziale su disco, che per diventare file programma dovrà passare tramite l'elaborazione di alcune linee di programma (490-504), che provvederanno a rileggere il file sequenziale, e tramite la solita tecnica del buffer della tastiera immettere in memoria linea per linea fino alla fine del file, dopodiché, le linee 506-516 provvederanno a calcolare il punto esatto dove termina il programma C64.DATAENTER, e tramite due poke nelle locazioni di memoria 43 e 44, (puntatore all'inizio del testo basic), si "cancellerà" il dataenter, per cui effettuando un SAVE, saranno salvate solo le linee basic aggiunte, contenente i data. A questo punto, sarà opportuno resettare i puntatori tramite uno spegnimento e riaccensione del computer oppure con il comando POKE43,1:POKE44,8:NEW <RETURN>.

Una ulteriore differenza nei listati delle due versioni del programma (oltre a quelle menzionate) è dato dalla presenza in coda alla versione del C64.DATAENTER di una routine in LM sotto forma di data, la quale, immessa a partire da $C000, si occupa totalmente dell'accettazione dei soli tasti previsti nelle tre differenti situazioni di input: input dei dati, input del checksum e correzione.

È stato infatti necessario ricorrere al LM per aumentare la velocità dell'immissione dei dati che altrimenti, a causa dei notevoli controlli e della ridefinizione dei tasti per creare il tastierino numerico, con il 'povero' basic 2.0 sarebbe divenuta esasperante. La stessa routine, si occupa pore dell'accensione e dello spegnimento del cursore ad ogni carattere prelevato. Un'altra sezione, ($C0FA $C168) è preposta alla creazione dei segnali sonori.

Chi volesse adattare il C64.DATAMAKER al funzionamento con registratore a cassette avrà ben poche modifiche da fare. Unico limite che si troverà usando questa periferica (oltre alla bassa velocità), sarà dato dal fatto che non potendo usufruire della funzione 'APPEND' nell'apertura di un file esistente per l'accodamento dei dati, il lavoro dovra' necessariamente essere eseguito in una unica seduta. A meno che...

Decesa.datastore

Questa non è altro che una semplice routinetta da aggiungere ai propri programmi, e che serve a leggere ed inserire in memoria i codici contenuti in linee data create con il programma DATAMAKER, di particolare ha che è universale, e cioè si adatta ad ognuno dei formati prodotti dai programmi MLX.DATAMAKER. I data potranno essere sia in dec che in esa, con il checksum sia a blocchi che totale. Il segnale di un checksum riguardante un blocco di linee, sarà dato dalla lettura di un solo asterisco, mentre alla lettura di due asterischi indicheranno alla routine che i dati sono terminati, e che quello che ci si appresta a leggere, sarà l'ultimo dato di controllo.

L'unica indicazione per l'uso della routine è quella di dover assegnare alla variabile "M" nella linea 15 il valore della locazione di memoria di inizio dei dati da immettere, mentre alla line 20 si dovrà assegnare il corretto valore alla variabile "FD" (flag): zero se i dati da leggere sono in esadecimale, oppure diverso da zero (uno) se si tratta di dati in base decimale.

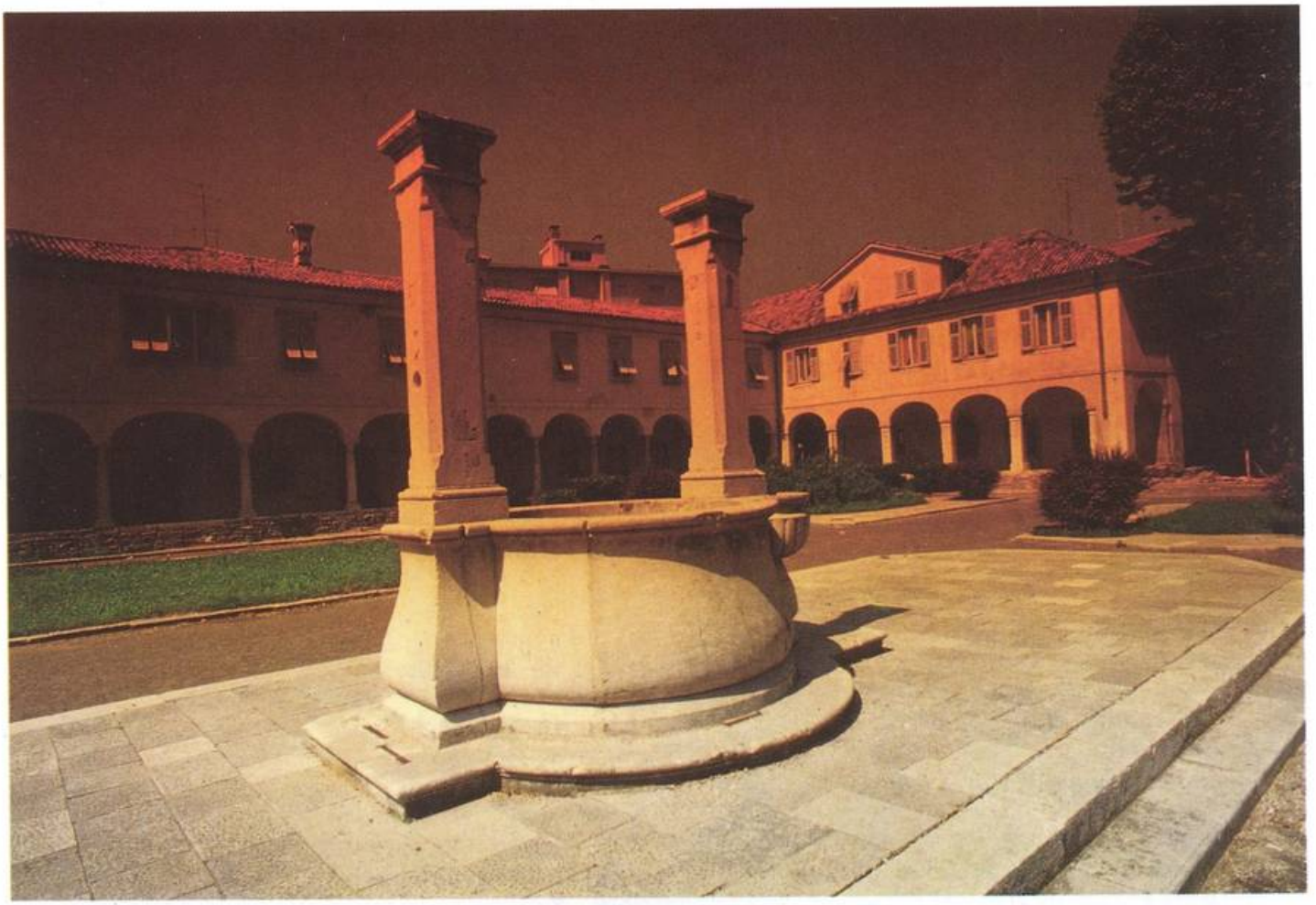
Conclusione

Ricordiamo che a causa dei continui riferimenti ai numeri di linea dentro stringhe o istruzioni che non sarebbero influenzate da un RENUMBER è consigliabile mantenere la numerazione così come viene riprodotta, nei programmi presentati con questo articolo. Raccomandando inoltre che per i due programmi per 128: MLX.TIMESTORE e DATAENTER, si abbia cura di copiare anche la prima linea contenente una rem seguita da alcuni asterischi. Speriamo con questo nostro ulteriore sforzo di incoraggiare anche i più restii fra i lettori di COMMODORE PROFESIONAL, i quali avrebbero voluto copiare i listati da noi presentati, ma che a causa delle difficoltà menzionate all'inizio, non se l'erano sentita. A tutti questi, nonché agli altri, auguriamo un buon lavoro con MLX e DATAENTER.

Per la gioia dei neofiti dell'ASSEMBLER pubblichiamo il disassemblato commentato della routine per l'input controllato e per l'emissione dei beep di controllo del programma c64.dataenter.

Siccome la routine è polivalente, può cioè accettare caratteri diversi in dipendenza dalla sezione di programma che la chiama, si sono usate le locazioni $FB $FC $FD $FE per comunicare con il programma Basic. Precisamente la locazione $FB, inpostata all'inizio del programma, se diversa da zero, comunica alla routine che l'input dei dati è in esadecimale, per cui saranno accettati anche i caratteri dalla "A" alla "F". Nella locazione $FC viene invece impostato il valore 0/2, a seconda della sezione di programma che chiama la routine: 0=input dati, 1=input checksum, 2=input correzione. In $FD viene invece depositato il carattere di ritorno, convertito nella cifra corrispondente se il tasto riguarda i caratteri che ridefiniscono il tastierino. Inoltre, per aumentare la velocità di elaborazione del programma basic nella locazione $FE, viene immesso un valore dipendente dal carattere di ritorno, ed in relazione del quale valore, si eseguirà da basic un salto ON GOTO.... eliminando così i lunghi confronti IF THEN....

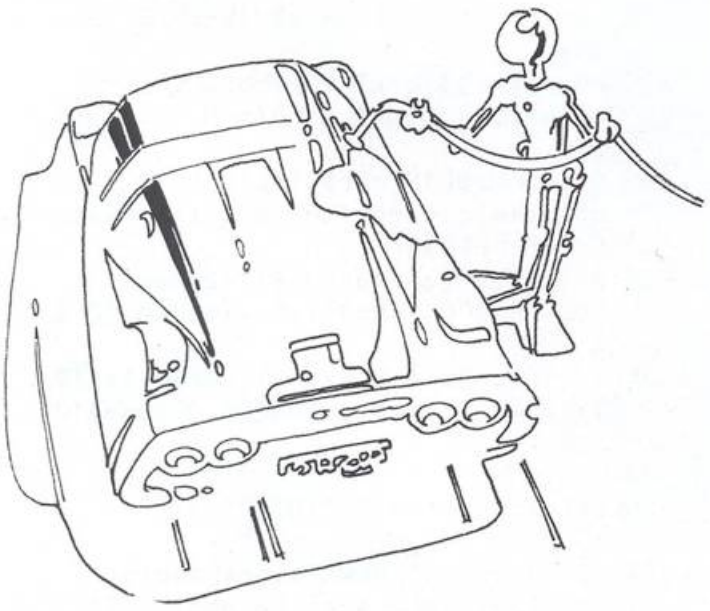
Il disassemblato è stato ottenuto con l'ottimo Macro Assembler per C64; ed essendo estremamente semplice, la routine non ha bisogno di ulteriori commenti.



128.dataenter

```
100 rem"*****************************
    ******************
102 data 24,d7,f0,04,20,21,cb,60,20,6
    f,cd,60,24,d7,30,06
104 data a9,80,20,b3,cd,60,20,9f,cd,6
    0,2a,2a,2a,2a,2a,2a
106 graphic clr:forj=1to32:reada$:
    pokedec("1c08")+j,dec(a$):next
108 :
110 goto116
112 sound1,40000,2,,,,1:return
114 :
116 rem 128.dataenter by a. de sabbat
    a
118 :
120 n$=chr$(144):vv$=chr$(30):ri$=
    chr$(157)
122 rv$=chr$(18):up$=chr$(145):gg$=
    chr$(158):rs$=chr$(31):dl$=chr$(2
    0)
124 le$=chr$(29):da$=rv$+"128.DATAENT
    ER[2 spc]by[spc]A.[spc]De[spc]Sab
    bata"
126 ca=dec("1c09"):cs=dec("1c15"):if
    rgr(0)=5thena%=80:fast:elsea%=40
128 x$=rv$:forc=1toa%:x$=x$+chr$(32):
    next
130 e$=up$+up$+chr$(27)+chr$(64):
    color6,1:color4,1:color0,1:color5
    ,15
132 printchr$(19)chr$(19)chr$(147)
    chr$(14);:forc=1to3:printx$:next:
    printchr$(19)
134 printspc((a%-len(da$))/2)da$:
    printrv$"(c)1988"vv$:window0,10,a
    %-1,24
136 fora=2to8step2:keya-1,chr$(64+a/2
    ):keya,"":next:poke4104,0:poke410
    5,0:vol10
138 :
140 print:print:m1$="10000"
142 :
144 a$="":print"[down]Linea[spc]di
    [spc]inizio[2 spc]:[3 spc]"m1$"
```

```
    [7 right]";:inputa$:ln=val(a$):if
    ln<1000orln>63900thengosub458:
    printe$;:goto144
146 :
148 b$="":input"[down]Dec[spc]o[spc]e
    sa[spc](d/e)[2 spc]:[3 spc]e
    [3 right]";b$:ifb$<>"d"andb$<>"e"
    thengosub458:printe$;:goto148
150 k0$="0123456789":ifb$="d"thenbb=1
    0:bz=3:md$="7":elsebb=16:bz=2:md$
    ="9":k0$=k0$+"abcdef+-"
152 dm=int((150-(40anda%=40))/(bz+1))
154 :
156 a$="":print"[down]Data[spc]per
    [spc]linea[3 spc]:[3 spc]"md$"
    [3 right]";:inputa$:dl=val(a$):if
    a$=""ordl<1ordl>dmthengosub458:
    printe$;:goto156:else in$=right$(
    "[spc]"+a$,2)
158 :
160 a$="":print"[down]Incremento
    [7 spc]:[3 spc]"in$"[4 right]";:
    inputa$:in=val(a$):ifa$=""orin<2
    or in>100 thengosub458:printe$;:
    goto160
162 a=dl:gosub474:ur$=uq$
164 :
166 a$="":input"[down]Cksum[spc]total
    e[spc]o[spc]a[spc]blocchi[spc](t/
    b):[3 spc]b[3 right]";a$:ifa$<>"t
    "anda$<>"b"thengosub458:printe$;:
    goto166:elsecc=0:ifa$="b"thencc=1
168 :
170 lx=cc:ifccthena$="":input"[down]N
    umero[spc]di[spc]linee[spc]per
    [spc]blocco[4 spc]:[3 spc]5
    [3 right]";a$:lb=val(a$):ifa$=""
    orlb<1orlb>lmthengosub458:printe$
    ;:goto170
172 :
174 k$="*"+dl$+k0$:ll=ln:dima(dl):
    scnclr:ifccthendiml$(lb)
176 bank(15)
178 :
180 rem --------- inizio loop -------
    ---
182 :
184 ck=0 : fd=0
186 j1=0
188 :
190 l$=right$("[3 spc]"+str$(ln),5)+"
    [spc]data[spc]"
192 printl$;:z=0:j=0:n=0
194 do
196 sysca:do:getkeya$:loopuntilinstr(
    k$,a$):syscs
198 ifa$="+"thena$="e":elseifa$="-"
    thena$="f"
200 a=asc(a$)
202 :
204 ifa$="*"thenbegin:if fd=0 then
    gosub458:goto196
206 :  ifz>0thenz=z-1:gosub316:goto20
    6
208 :  ifz=0andj=0thenprint:ln=ln-in:
    printup$chr$(27)chr$(81);:goto212
210 :  gosub316:print
212 :  je=1:lb=j1+1:exit
214 bend
216 :
218 ifa=20 then begin
220 :  ifz=0andj=0thengosub458:goto19
    6
222 :     ifz>0thenbegin:gosub316:z=z
    -1:ifz=0thenn=0:gosub112:goto196
224 :       ifz=1thenn=int(n/(bb*bb))
    *bb*bb:gosub112:goto196
226 :     bend
228 :  ifz=0andj>0thengosub316:gosub3
    16:z=bz-1:j=j-1:fd=fd-1:n=int(a(j
    )/bb)*bb
230 gosub112:goto196:bend
232 :
234 n1=a-(48+(7anda>57)):onz+1+(1andb
    z=2)goto236,238,240
236 n2=n+bb*bb*n1:goto242
238 n2=n+bb*n1  :goto242
240 n2=n+n1
242 ifn2>255then:gosub458:goto196:
```



```
    elsen=n2
244 l$=l$+a$:printa$;:gosub112:z=z+1:
    ifz<bz then196
246 a(j)=n:n=0:z=0:j=j+1:fd=fd+1:ifj<
    >dlthenprint",";:l$=l$+",":else
    print
248 loopwhile j < dl
250 :
252 ln=ln+in:if je=1 and j>0 then j=j
    +1:elseifje=1then266
254 l$(j1)=l$:for i=0 to (j-1)-(1andj
    e=1):ck=ck+a(i):next
256 gosub466:if cc then j1=j1+1:else4
    16
258 if j1<lb then190
260 :
262 rem -------- input ceksum -------
    ---
264 :
266 n=0:z=0: ln=(ln-in)+1
268 l$=right$("[3 spc]"+str$(ln),5)+"
    [spc]data[spc]*,":printl$;:ifje=1
    thenje=0:cc=1:goto278
270 sysca:do:getkeya$:loopuntilinstr(
    k0$+chr$(13)+dl$,a$):syscs
272 ifa$="+"thena$="e":elseifa$="-"
    thena$="f"
274 :
276 ifa$="*"thenbegin
278 : ifje=0thengosub316:print"*,";:l
    $=l$+"*,":je=1:elsegosub458
280 goto270:bend
282 :
284 ifa$=dl$thenbegin
286 : ifz>0thengosub316:n=int(n/16):z
    =z-1:gosub112:goto270:elsegosub45
    8:goto270
288 bend
290 :
292 ifa$=chr$(13)thenbegin:ifz=0then
    gosub458:goto270:elseprint
294 : la$=left$(l$,14-(1andje=0)):lb$
    =mid$(l$,15-(1andje=0))
296 : lb$=right$("0000"+lb$,4+(1andbb
    =10)):l$(j1)=la$+lb$
298 : ln=ln+(in-1):ifn<>ckthengosub45
    8:goto324:elsegosub466:goto416
300 bend
302 :
304 ifz>3+(1andbb=10)thengosub458:
    goto270:elsea=asc(a$)
306 z=z+1:l$=l$+a$:printa$;:n=n*bb+a-
    (48+(7anda>57)):gosub112:goto270
308 :
310 rem --------- subroutine --------
    ---
312 rem elimina l'ultimo carattere da
     l$
314 :
316 l$=left$(l$,len(l$)-1):printdl$;:
    return
318 :
320 rem ---------- correzione -------
    ---
322 :
324 if je=1 and lx=0 then fr=1:l$(0)=
    l$(0)+":rem[spc]ceksum[spc]errato
    !":goto416
326 printgg$;:lk=lb:ifje=1andj=0andcc
    =1thenlk=lb-1:goto330
328 ifje=1thena=j-1:gosub474
330 gosub484:for c=0 to lk:printl$(c)
    :next
332 gosub484
334 :
336 jc=0:b=bz-1:dd=dl-1:s=12
338 z=0:d=0:printspc(s-1);
340 sysca:do:getkeya$:loopuntilinstr(
    k0$+"[left][right]"+chr$(13),a$):
    syscs
342 ifa$="+"thena$="e":elseifa$="-"
    thena$="f"
344 :
346 ifa$=chr$(13)thenbegin:print
348 :  if jc=lk thenbegin:a=14:if je=
    1 thena=15
350 :     fast:ifbb=16thenck=dec(mid$
    (l$(jc),a)):elseck=val(mid$(l$(jc
    ),a))
352 :     n=0:fori=0tojc-1
354 :     yy=len(l$(i))-11:yy=int(yy/
    (bz+1)):fory=0to yy
356 :     ifbz=3thena=val(mid$(l$(i),
    12+4*y,3)):elsea=dec(mid$(l$(i),1
    2+3*y,2))
358 :     n=n+a:nexty,i:ifa%=40then
    slow
360 :     ifck<>nthengosub458:goto332
    :elsegosub466:goto416
362 :  bend
364 :  jc=jc+1:if jc<lk thengosub466:
    goto338:elsebegin
366 :    dd=0:b=len(l$(jc))-14:s=14:a
    $=mid$(l$(jc),13,1):ifa$="*"thens
    =15:b=b-1
368 :    gosub466:goto338
370 :  bend
372 bend
374 :
376 ifa$=le$thenbegin
```

```
378 : if pos(0)=len(l$(jc))-1 then
    gosub458:goto340
380 : printa$;:gosub112
382 : z=z+1:ifz=b+1 thenz=0:d=d+1:
    printle$;
384 goto340:bend
386 :
388 ifa$=ri$thenbegin
390 : ifz=0 and d=0 thengosub458:goto
    340
392 : z=z-1:ifz<0 thenz=b:d=d-1:print
    ri$;
394 : printri$;:gosub112
396 goto340:bend
398 :
400 printa$;:gosub112
402 mid$(l$(jc),s+(b+2)*d+z,1)=a$
404 if pos(0)=len(l$(jc)) thenprintri
    $;:goto408
406 z=z+1:ifz=b+1thenz=0:d=d+1:printl
    e$;
408 goto340
410 :
412 rem -- append lines to basic prog
    ram --
414 :
416 ifccthenam=lb-(1andje=1andj=0):
    elseam=0
418 lk=am:gosub484:for c=0 to am:
    printrs$l$(c):next
420 bank15:y=peek(235)
422 window 0,3,a%-1,9,1:aj=0
424 :
426 printchr$(147)rs$l$(aj):printn$"g
    oto430"
428 poke208,3:poke842,19:poke843,13:
    poke844,13:end
430 aj=aj+1:if aj =< am then426
432 :
434 scnclr:printvv$:window 0,10,a%-1,
    24:sysdec("cc6a"),0,y-10,0
436 if je=0 and cc=1 then184:elseif j
    e=0 and cc=0 then186
438 if fr=1 then446:else if je=1 and
    cc=1 then448
440 if cc=0 and je=1 then266
442 :
444 rem ------ esce o ritorna ------
446 printgg$chr$(17)rv$"[10 spc]E
    [spc]R[spc]R[spc]O[spc]R[spc]E
    [10 spc]"
448 printvv$chr$(17)chr$(17)"delete
    [spc]-"11-1:y=peek(235)-3
450 window0,3,a%-1,24:sysdec("cc6a"),
    0,y-3,0:end
452 :
454 rem -------- beep errore --------
456 :
458 sound1,5000,15,,,,1:return
460 :
462 rem --------- beep ok! ----------
464 :
466 sound1,12000,5,,,,1:return
468 :
470 rem ------ definisce uq$ --------
472 :
474 uq$="":a1=(a*(bz+1)+10)/a%:a2=int
    (a1):ifa1>a2thena2=a2+1
476 lm=int(13/a2):fori=1toa2:uq$=uq$+
    up$:next:return
478 :
480 rem -- posiziona cursore sul prim
    o dato --
482 :
484 printup$uq$;:c=1
486 ifc<lkthenprintur$;:c=c+1:goto486
488 return
10000 data 00,01,00,00,00,00,00,00,00
10009 data 00,00,00,00,00,00,00,00,00
10010 data *,0001
```

128mlx.timestore

```
    0 rem"*****************************
      **********
    1 data 24,d7,f0,04,20,21,cb,60,20,6
      f,cd,60,24
    2 data d7,30,06,a9,80,20,b3,cd,60,2
      0,9f,cd,60
    3 :
  100 rem [rvon]-                  mlx 1
      28  a. de sabbata
                    -
  102 data145,27,81,145,145,13
  104 color6,1:color4,1:color0,1:color5
      ,15:a%=39:ifrgr(0)=5thena%=79:
      fast
  106 gg$=chr$(158):rs$=chr$(150):pr$=
      chr$(129):dl$=chr$(20):vol10
  108 dw$=chr$(17)+chr$(17):r$=chr$(18)
  110 ee$=chr$(27)+"q":e$="[2 up]"+chr$
      (27)+"`":da$=r$+"128.MLX[2 spc]by
      [spc]A.[spc]De[spc]Sabbata"
  112 pudef"0":u$="###,":u1$="#####
```

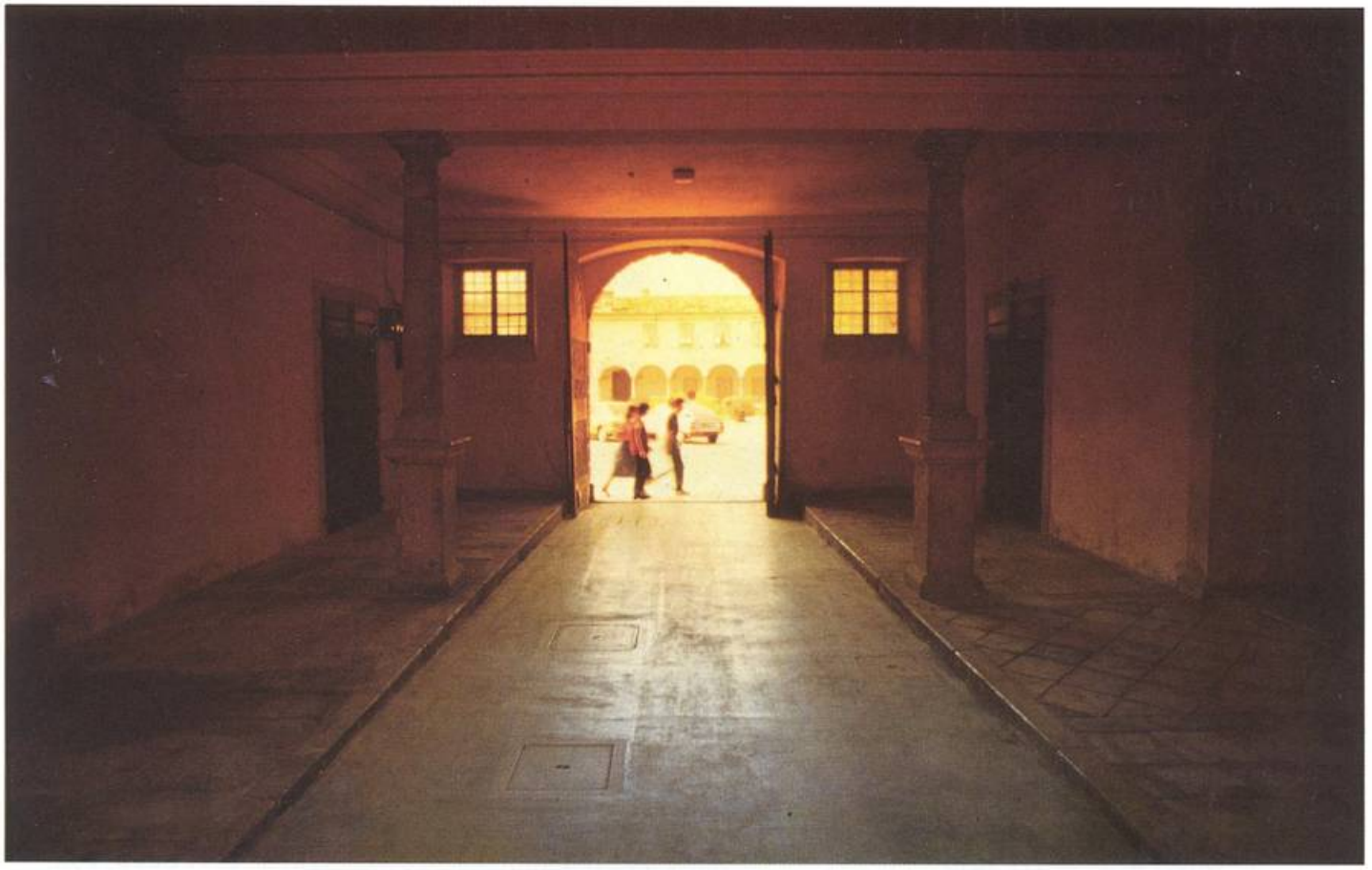
```
    [spc]:":vv$=chr$(30)
114 x$=r$:forj=0toa%:x$=x$+chr$(32):
    next
116 printchr$(19)chr$(19)chr$(147)
    chr$(14);:forj=1to3:printx$:next:
    printchr$(19)
118 printspc((a%-len(da$))/2)da$:
    printr$"(c)1988"vv$:window0,4,a%,
    24
120 bank(15)
122 ifpeek(254)=128thenbegin:poke254,
    0
124 : forj=1topeek(4864):f$=f$+chr$(
    peek(4864+j)):next
126 : s=peek(250)*256+peek(251)
128 : e=peek(252)*256+peek(253)
130 : sm=peek(143):bb=peek(142):bank(
    bb)
132 goto242:bend
134 forj=1to8:keyj,"":next:poke4104,0
    :poke4105,0
136 :
138 rem [rvon]-
     input parametri
                    -
140 print
142 a$="":print"[spc]Numero[spc]del
    [spc]banco[spc](0/1)[5 spc]";:
    inputa$:bb=val(a$)
144 ifa$=""orbb<0orbb>1thengosub440:
    printe$:goto142
146 bank(bb)
148 :
150 print:print:goto154
152 gosub440:printe$
154 f$="":input"[spc]Nome[spc]del
    [spc]file[spc](max[spc]14[spc]ca
    .)";f$:iff$=""then152
156 f$=left$(f$,14)
158 :
160 print:print:goto164
162 gosub440:printe$
164 a$="":print"[spc]Step[spc]minuti
    [spc]per[spc]save[spc](1-30)";:
    inputa$:sm=val(a$):ifa$=""then162
166 ifsm<1orsm>30then162
168 :
170 print:print:goto174
172 gosub440:printe$
174 a$="":print"[spc]Numero[spc]del
    [spc]file[spc](0/99)[5 spc]";:
    inputa$:nf=val(a$):ifa$=""then172
176 ifnf<0ornf>99then172:elsef$=
    right$("0"+mid$(str$(nf),2),2)+f$
178 :
180 print:print:goto184
182 gosub440:printe$
184 a$="":print"[spc]Indirizzo[spc]in
    iziale[spc](dec.)[2 spc]";:inputa
    $:s=val(a$):ifa$=""then182
186 ifbb=1ands<dec("0400") or s>dec("
    fefa")then182
188 ifbb=0thenbegin
190 : ifs<dec("0b00")or s>dec("fefa")
    then182
192 : ifs>dec("0ffa")ands<dec("1300")
    then182
194 bend
196 :
198 print:print:goto202
200 gosub440:printe$
202 a$="":print"[spc]Indirizzo[spc]co
    nclusivo[spc](dec.)";:inputa$:e=
    val(a$):ifa$=""then200
204 ifs=e or e<s or e>dec("fefe")then
    200
206 ifbb=1thenbegin:bank(0)
208 : ifs<dec("8000")thenbegin:gosub4
    52:print"[clr][down]Modifico[spc]
    la[spc]base[spc]della[spc]memoria
    [down]
210 :   print"poke48,"he+1":poke47,"l
    e":clr:run":sleep1:goto466
212 : bend
214 : ife>dec("8000")thenbegin:gosub4
    52:print"[clr][down]Modifico[spc]
    la[spc]cima[spc]della[spc]memoria
    [down]
216 :   print"poke54,"hs-1":poke53,"l
    s":clr:run":sleep1:goto466
218 : bend
220 bend
222 ifbb=0thenbegin
224 :  if s< dec("1000")ande>dec("0ff
    f")then200
226 :  if(s=>dec("1300")ands<dec("1c0
    0")ande>=dec("1c00"))or(s>=dec("1
    c00")ands<dec("3400"))thenbegin
228 :     printr$"[clr][spc]Rilocazio
    ne[spc]MLX:[spc]"dw$:gosub452
230 :     h$=mid$(str$(int((e+2)/256)
    ),2):l$=mid$(str$((e+2)-val(h$)*2
    56),2)
232 :     print"poke46,"h$":poke45,"l
    $":poke"mid$(str$(e+1),2);
234 :     print",0:clr:run"chr$(34)"1
    28mlx.time*"+chr$(34):goto466
236 :  bend
238 bend
240 :
```

```
242 graphic clr:bank(15):a=(peek(45)+
    peek(46)*256)+7
244 forj=0to25:reada$:poke a+j,dec(a$
    ):next
246 ca=a:cs=a+12:ad=s:scnclr:print:ti
    $="000000":tm=sm*3600
248 :
250 rem [rvon]-
    inizio programma
                    -
252 printvv$usingu1$;ad;:forj=1to7:
    bank(15):gosub282
254 ifa>128thenbegin
256 : a=a-194:a=a-(7anda=10)-(8anda=1
    2)-(12anda=17)
258 : onagosub372,400,356,386,334:
    goto252
260 bend
262 :
264 a(j)=n1:nextj
266 ck=ad-int(ad/256)*256:fori=1to6:c
    k=(ck+a(i))and255:next
268 printdl$:ifck<>n1thengosub436:
    goto252
270 gosub444:bank(bb)
272 fori=1to6:pokead+i-1,a(i):next
274 if(ti>tm)and(peek(211)<>16)then
    sound1,10000,30,2,8000,2000:gosub
    334
276 ad=ad+6:ifad=<ethen252:else334
278 :
280 rem [rvon]-                    ro
    utine di input  -
282 z=0:n1=0:n=0
284 printchr$(30+(128andj=7))chr$(18+
    (128andj<7));:a$=""
286 sysca:do:getkeya$:loopuntilinstr(
    "0123456789CDLNS"+chr$(20),a$):
    syscs
288 a=asc(a$):sound1,36000,1,,,,0
290 ifa>128thenreturn
292 :
294 ifa=20thenbegin
296 :  ifz=0andj=1thengosub440:goto28
    2
298 :  ifz>0thenbegin:printdl$;:z=z-1
    :ifz=0thenn1=0:goto284
300 :     n1=int(n1/(100))*100:goto28
    4
302 :  bend
304 :  ifz=0andj>1thenprintdl$;:print
    dl$;:z=2:n=-1:j=j-1:n1=int(a(j)/1
    0)*10
306 :  printchr$(30+(128andj=7))chr$(
    18+(128andj<7));
308 :  goto284
310 bend
312 n2=a-48:onz+1goto314,316,318
314 n3=n2*100:goto320
316 n3=n1+n2*10 :goto320
318 n3=n1+n2
320 ifn3>255thengosub440:goto284:else
    n1=n3
322 printa$;
324 z=z+1:ifz<3then284
326 print",";
328 return
330 :
332 rem [rvon]-
        save
                    -
334 print:printdw$e$vv$r$"[up][spc]>E
    SCAPE<[spc]per[spc]uscire[spc]dal
    [spc]save[spc]"
336 gosub472
338 ifa$=chr$(27)and ad <= e then
    print:printe$;:return
340 nf=nf+1:ifnf>99thennf=0
342 f$=right$("0"+mid$(str$(nf),2),2)
    +mid$(f$,3)
344 scratch(f$):ifds>1thengosub440:
    goto348
346 bsave(f$),b(bb),p(s) to p(e)
348 printds$:ifdsthengosub472:nf=nf-1
    +(100andnf=0):elsesleep1:ti$="000
    000"
350 ifad=<ethenprinte$;:return:else
    new:end
352 :
354 rem [rvon]-
        load
                    -
356 print:printdw$e$;:printr$"[up]
    [spc]>ESCAPE<[spc]per[spc]uscire
    [spc]dal[spc]load[spc]"
358 gosub472:ifa$=chr$(27)thenprint:
    printe$;:return
360 dopen#1,(f$):close1:ifdsthengosub
    440:goto364
362 bload(f$),b(bb)
364 printds$:ifdsthengosub472:else
    sleep1
366 printe$;:return
368 :
370 rem [rvon]-
    catalog
372 scnclr:printgg$;:catalog
374 :  ifdsthenprint:printr$ds$:gosub
    440:sleep1:scnclr:print:return
```

```
376 :  printr$"[spc]Un[spc]tasto[spc]
    per[spc]continuare..."
378 :  gosub472:scnclr:print
380 return
382 :
384 rem [rvon]-                    nu
    ovo indirizzo
                -
386 printpr$
388 :  a$="":input"[up]Nuovo[spc]indi
    rizzo[spc]";a$:zz=val(a$)
390 :  ifa$=""thenprint:printe$"[up]"
    :goto252
392 :  ifzz<sorzz>ethengosub438:sleep
    1:print:printe$:goto388
394 :  ad=zz:print:printe$;
396 goto252
398 rem [rvon]-
      display memoria
                -
400 printgg$
402 :  a$="":input"[up]Listato[spc]da
    [spc]";a$:f=val(a$)
404 :  ifa$=""thenprint:printe$;:
    return
406 :  a$="":printtab(20)"[up]a[spc]"
    ;:inputa$:t=val(a$)
408 :  ifa$=""thenprint:printe$;:
    return
410 :  iff<sorf>eort<sort>ethengosub4
    34:sleep1:print"[3 up]"ee$:goto40
    2
412 :  bank(bb):print:printe$"[up]";
414 :  fori=ftotstep6:print:print
    usingu1$;i;
416 :    fork=0to5:n=peek(i+k):print
    usingu$;n;:next:printdl$;
418 :  geta$:ifa$>""thenprint:return
420 : nexti:print
422 return
424 :
432 rem [rvon]-
      beep errore
                -
434 print"[down]Minimo";s;"[spc]massi
    mo";e:goto440
436 printrs$r$"[11 spc]E[spc]R[spc]R
    [spc]O[spc]R[spc]E[12 spc]":goto4
    40
438 printr$"[up][2 spc]Esce[spc]dal
    [spc]range[spc]consentito![2 spc]
    "
440 sound1,5000,20,,,,1:return
442 :
444 rem [rvon]-
      beep ok!
446 sound1,12000,5,,,,1:return
448 :
450 rem [rvon]-                salva pa
    rametri per rilocazione mlx
                -
452 bank(15):poke4864,len(f$):forj=1
    tolen(f$):poke4864+j,asc(mid$(f$,
    j,1)):next
454 hs=int(s/256):ls=s-hs*256:poke250
    ,hs:poke251,ls
456 he=int(e/256):le=e-he*256:poke252
    ,he:poke253,le
458 poke254,128:poke143,sm:poke142,bb
460 return
462 :
464 rem [rvon]-
      poke in buffer
                -
466 restore102:forj=1to6:reada:poke84
    1+j,a:next:poke208,6:end
468 :
470 rem [rvon]-                    as
    petta un tasto qualsiasi
                -
472 bank(15):a$="":sysca:getkeya$:sys
    cs:bank(bb):return
```

128mlx.datamaker

```
  0 rem [rvon]---------------- mlx da
    tamaker    by  a. de sabbata ----
    ------------------
  1 p$=chr$(144):g$=chr$(30):e$="
    [2 up]"+chr$(27)+"`":r$=chr$(18):
    da$=r$+"MLX.DATAMAKER[2 spc]by
    [spc]A.[spc]De[spc]Sabbata":color
    6,1:color4,1:color0,1:a=40:ifrgr(
    0)=5thena=80:fast
  2 window0,0,a-1,24,1:color5,15:x$=r
    $:forj=1toa:x$=x$+"[spc]":next:
    forj=1to3:printx$:next:printchr$(
    19)chr$(14):printspc((a-len(da$))
    /2)da$g$:window0,4,a-1,24,1
  3 input"[down]Formato[spc]MLX[spc](
    s/n):[3 spc]n[3 right]";mx$:ifmx$
    <>"s"andmx$<>"n"thenprinte$;:goto
    3:elsemx=0:ifmx$="s"thenmx=1
  4 input"[down]locaz.[spc]iniziale
    [2 spc]:[spc]";m1:ifm1<dec("0400"
    )thenprinte$;:goto4:elsem1$=
```



```
   right$("[3 spc]"+str$(m1),5)
 5 input"[down]locazione[spc]finale
   [spc]:[spc]";m2:ifm2=<m1orm2>dec(
   "feff")thenprinte$;:goto5
 6 input"[down]numero[spc]banco
   [5 spc]:[3 spc]0[3 right]";nb:ifn
   b<0ornb>1thenprinte$;:goto6
 7 ifmxthenln=m1:goto30
 8 input"[down]dec[spc]o[spc]esa
   [spc](d/e)[2 spc]:[3 spc]e
   [3 right]";b$:ifb$<>"d"andb$<>"e"
   thenprinte$;:goto8:elsedm$="07":b
   z=3:ifb$="e"thendm$="10":bz=2
 9 print"[down]linea[spc]di[spc]iniz
   io[2 spc]:[3 spc]"m1$"[7 right]";
   :inputln:ifln<50thenprinte$;:goto
   9
10 dm=int((150-(40anda=40))/(bz+1)):
   print"[down]data[spc]per[spc]line
   a[3 spc]:[3 spc]"dm$"[4 right]";:
   inputdl:ifdl<1ordl>dmthenprinte$;
   :goto10:elsein$=mid$(str$(dl),2)
11 print"[down]incremento[7 spc]:
   [3 spc]"in$"[4 right]";:inputin:
   ifin<2or ln+(m2-m1)/dl*in>63999
   thenprinte$;:goto11
12 input"[down]cksum[spc]totale[spc]
   o[spc]a[spc]blocchi[spc](t/b):
   [3 spc]b[3 right]";c$:ifc$<>"t"
   andc$<>"b"thenprinte$;:goto12:
   elsecc=0:lb=-1:ifc$="b"thencc=1
13 ifcctheninput"[down]numero[spc]di
   [spc]linee[spc]per[spc]blocco
   [4 spc]:[3 spc]05[4 right]";lb:if
   lb<1orlb>20thenprinte$;:goto13
14 rem [rvon]-------------- costruzi
   one linee data formato cksum ----
   ---------------
15 bank(nb):ii=0:pudef"0":ck=0:l1=ln
   :l1=-1:ifb$="d"thenfd=1:elsefd=0
16 l1=l1+1:printchr$(147)g$;ln-((in-
   1)andl1=lb)"data[spc]";:rem ** st
   art **
17 ifccandl1=lbthenbegin:print"*,";
18 : iffdthenprintmid$(str$(ck),2):
   elseprinthex$(ck)
19 : printp$"[spc]goto16":ck=0:l1=-1
   :goto39
20 bend
21 a=peek(m1):iffdthenprintusing"###
   ";a;:elseprintright$(hex$(a),2);
22 ii=ii+1:m1=m1+1:ck=ck+a
23 ifm1>m2thenbegin:print:println+1"
   data[spc]**,";
```

```
24 : iffdthenprintmid$(str$(ck),2):
   elseprinthex$(ck)
25 : print"[down]delete[spc]-"ll-1:
   goto39
26 bend
27 ifii<dlthenprint",";:goto21:else
   print:printp$"[spc]goto16"
28 goto38
29 rem [rvon]-------------- costruzi
   one linee data formato mlx ------
   ---------------
30 bank(nb):ii=0:pudef"0":ck=0:ll=ln
   :dl=6:in=6:n=0:dim v(5)
31 printchr$(147)g$;ln":";:v=ln-int(
   ln/256)*256:rem ** start **
32 a=peek(m1):a$=right$("00"+mid$(
   str$(a),2),3):printusing"###,";a;
33 forj=1to3:b=asc(mid$(a$,j,1)):n=n
   *10+b-48:next:v(ii)=n:n=0
34 ii=ii+1:m1=m1+1:ifii<dl then32:
   elsebegin
35 : ck=v:forj=0to5:ck=(ck+v(j))and2
   55:next:printusing"###";ck
36 : ifm1=<m2thenprintp$"[spc]goto31
   ":elseprint:print"[down]delete
   [spc]-"ll-1
37 bend
38 ln=ln+in
39 ii=0:poke208,4:poke842,19:poke843
   ,13:poke844,13:poke845,13:end
40 rem [rvon]----------------------
   ---------------------------------
   ---------------
```

decesa.datastore

```
   10 ifpeek(215)thenfast
   15 m=dec("3000"):rem locazione inizi
      o memoria
   20 fd=00          :rem flag (fd=00 dat
      a in esa, fd=01 data in dec.)
   25 :
   30 goto50
   35 reada$
   40 iffdthena=val(a$):elsea=dec(a$):
      return
   45 :
   50 do:reada$:ifa$="**"thenexit
   55 :  ifflag=0thenl=(peek(66)*256+
      peek(65)):flag=1
   60 :  ifa$="*"thenbegin:flag=0:gosub
      35
   65 :  :  ifck<>athenprint"[ctrl<g>]e
      rrore[spc]nelle[spc]linee"l"-"
      peek(66)*256+peek(65):end
   70 :  bend:ck=0
   75 :  gosub40:ifa>255thenprint"
      [ctrl<g>]errore[spc]nella[spc]lin
      ea"peek(66)*256+peek(65):end
   80 :  pokem,a:m=m+1:ck=ck+a
   85 loop
   90 gosub35:ifck<>athen65
   95 :
```

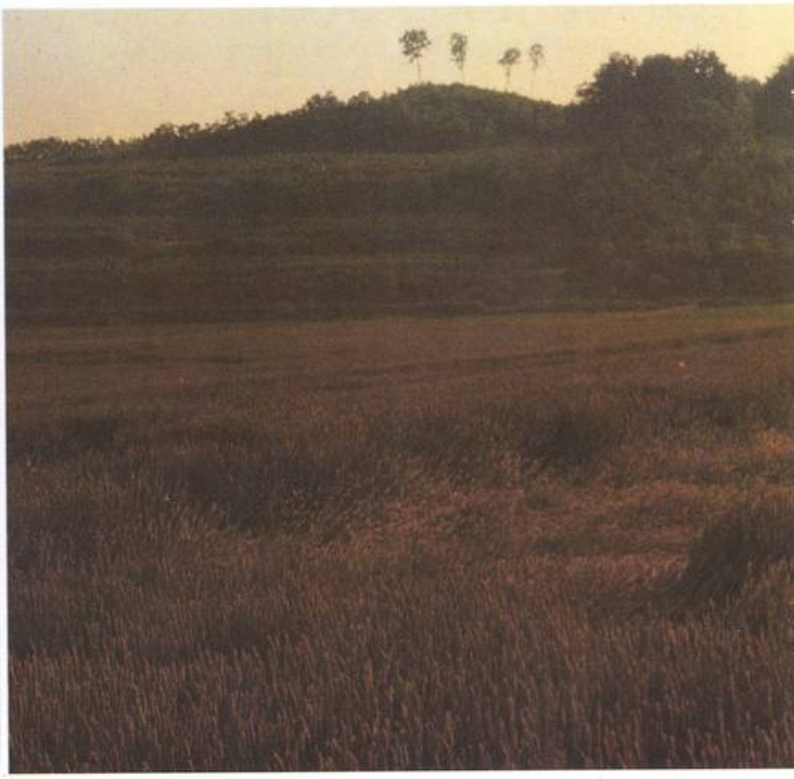
c64.dataenter

```
  100 rem  c64.dataenter by a. de sabba
      ta
  101 ber=49402:bok=49425:bip=49448
  102 n$=chr$(144):vv$=chr$(30):ri$=
      chr$(157)
  104 rv$=chr$(18):up$=chr$(145):gg$=
      chr$(158):rs$=chr$(31):dl$=chr$(2
      0)
  106 le$=chr$(29):da$=rv$+"C64.DATAENT
      ER[2 spc]by[spc]À.[spc]De[spc]Sab
      bata"
  108 a%=40:print"[clr][ctrl<n>]
      [ctrl<8>]";
  110 x$=rv$:forc=1toa%:x$=x$+chr$(32):
      next
  112 e$="[up]"+mid$(x$,3)+"[2 up]":
      poke53280,0:poke53281,0
  114 ifpeek(49154)<>133orpeek(49155)<>
      204thenprint"Attendere...":gosub5
      72
  116 print"[clr]";:forc=1to3:printx$;:
      next:print"[home][down]";
  118 printspc((a%-len(da$))/2)da$rv$"
      [home][2 down](c)1988"vv$
  120 :
  122 print:print:m1$="10000"
  124 :
  126 a$="":print"[down]Linea[spc]di
      [spc]inizio[2 spc]:[3 spc]"m1$"
      [7 right]";:inputa$
  128 ln=val(a$):ifln<1000orln>63900
      thensysber:printe$:goto126
  130 :
  132 b$="":input"[down]Dec[spc]o[spc]e
      sa[spc](d/e)[2 spc]:[3 spc]e
      [3 right]";b$
  134 ifb$<>"d"andb$<>"e"thensysber:
      printe$:goto132
  136 ifb$="d"thenbb=10:bz=3:md$="?":
```

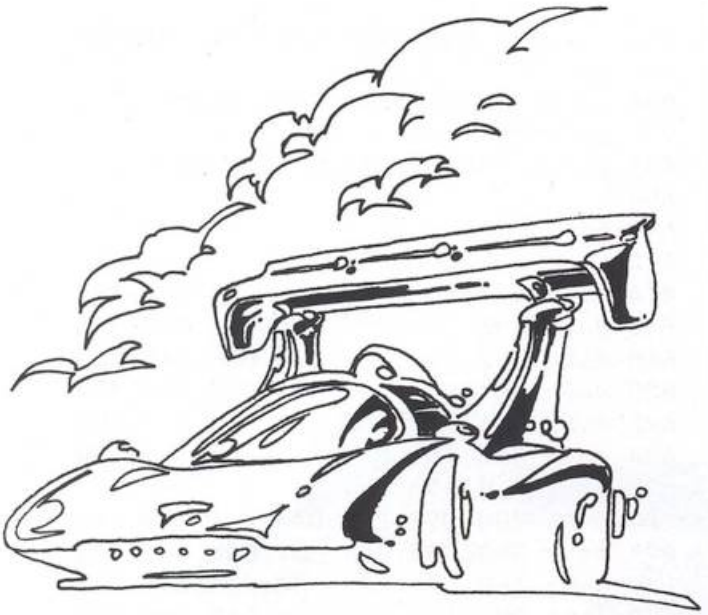
```
    poke252,0:goto142
138 bb=16:bz=2:md$="9":poke252,1
140 deffn de(x)=x-48+(x>57)*7
142 dm=int(70/(bz+1)):rem max data pe
    r linea
144 :
146 a$="":print"[down]Data[spc]per
    [spc]linea[3 spc]:[3 spc]"md$"
    [3 right]";:inputa$:in$=right$("
    [spc]"+a$,2)
148 dl=val(a$):ifa$=""ordl<1ordl>dm
    thensysber:printe$:goto146
150 :
152 a$="":print"[down]Incremento
    [7 spc]:[3 spc]"in$"[4 right]";:
    inputa$:in=val(a$)
154 ifin<2or in>100 thengsysber:print
    e$:goto152
156 a1=(dl*(bz+1)+10)/a%:a2=int(a1):
    ifa1>a2thena2=a2+1
158 lm=int(13/a2):fori=1toa2:uq$=uq$+
    up$:next
160 :
162 a$="":input"[down]Cksum[spc]total
    e[spc]o[spc]a[spc]blocchi[spc](t/
    b):[3 spc]b[3 right]";a$
164 ifa$<>"t"anda$<>"b"thensysber:
    printe$:goto162
166 cc=1:ifa$="t"thencc=0:goto174
168 :
170 lx=cc:ifccthena$="":input"[down]N
    umero[spc]di[spc]linee[spc]per
    [spc]blocco[4 spc]:[3 spc]5
    [3 right]";a$
172 lb=val(a$):ifa$=""orlb<1orlb>lm
    thensysber:printe$:goto170
174 ll=ln:dima(dl):ifccthendiml$(lb)
176 :
178 input"[down]Nome[spc]file[spc]su
    [spc]disco[spc]:[spc]";fi$:fi$=
    left$(fi$,16)
180 iffi$=""thensysber:printe$:goto17
    8
182 open1,8,15,"i0":gosub542
184 open2,8,2,(fi$+",s,w"):gosub542
186 ife1=63thenclose2:open2,8,2,(fi$+
    ",s,a"):gosub542
188 a=dl:gosub532:ur$=uq$:print"[clr]
    ":poke650,128
190 :
192 ck=0 : fd=0
194 j1=0
196 l$=right$("[3 spc]"+str$(ln),5)+"
    [spc]data[spc]"
198 printl$;:z=0:j=0:n=0
200 :
202 rem ------- inizio loop ---------
    -
204 :
206 poke251,0:rem preleva un car. in
    a$
208 sys49152:a=peek(253):a$=chr$(a)
210 onpeek(254)goto246,216,230,486
212 :
214 rem * asterisco
216 if fd=0 thensysber:goto208
218 ifz>0thenz=z-1:gosub340:goto218
220 ifz=0andj=0thenprint:print:ln=ln-
    in:printe$:goto224
222 gosub340:print
224 je=1:lb=j1+1:goto268
226 :
228 rem delete
230 ifz=0andj=0thensysber:goto208
232 : ifz=0then238
234 :   gosub340:z=z-1:ifz=0thenn=0:
    sysbip:goto208
236 :   ifz=1thenn=int(n/(bb*bb))*bb*
    bb:sysbip:goto208
238 : ifz=0andj>0thengosub340:gosub34
    0:z=bz-1:j=j-1:fd=fd-1:n=int(a(j)
    /bb)*bb
240 sysbip:goto208
242 :
244 rem carattere
246 n1=a-(48+(7anda>57)):onz+1+(1andb
    z=2)goto248,250,252
248 n2=n+bb*bb*n1:goto254
250 n2=n+bb*n1:goto254
252 n2=n+n1
254 ifn2>255thensysber:goto208
256 n=n2
258 l$=l$+a$:printa$;:sysbip:z=z+1:if
    z<bz then208
260 a(j)=n:n=0:z=0:j=j+1:fd=fd+1:ifj<
    >dlthenprint",";:l$=l$+",":goto26
    4
262 print
264 if j < dl then208
266 :
268 ln=ln+in:if je=1 and j>0 then j=j
    +1
270 ifj=0then284 : rem verificare!!!
272 l$(j1)=l$:for i=0 to (j-1)-(1andj
    e=1):ck=ck+a(i):next
274 sysbok:if cc=0 then452
276 j1=j1+1:if j1<lb then196
278 :
280 rem -------- input ceksum -------
    --
```

```
282 :
284 poke251,1:n=0:z=0: ln=(ln-in)+1
286 l$=right$("[3 spc]"+str$(ln),5)+"
    [spc]data[spc]*,":printl$;:ifje=1
    thenje=0:cc=1:goto298
288 :
290 sys49152:a=peek(253):a$=chr$(a)
292 onpeek(254)goto328,314,306,298
294 :
296 rem * asterisco
298 ifje=0thengosub340:print"*,";:l$=
    l$+"*,":je=1:goto290
300 sysber:goto290
302 :
304 rem delete
306 ifz>0thengosub340:n=int(n/bb):z=z
    -1:sysbip:goto290
308 sysber:goto290
310 :
312 rem return
314 ifz=0thensysber:goto290
316 print:la$=left$(l$,14-(1andje=0))
    :lb$=mid$(l$,15-(1andje=0))
318 lb$=right$("0000"+lb$,4+(1andbb=1
    0)):l$(j1)=la$+lb$
320 ln=ln+(in-1):ifn<>ckthensysber:
    goto348
322 lk=lb:sysbok:goto452
324 :
326 rem carattere
328 ifz>3+(1andbb=10)thensysber:goto2
    90
330 z=z+1:l$=l$+a$:printa$;:n=n*bb+a-
    (48+(?anda>57)):sysbip:goto290
332 :
334 rem --------- subroutine --------
    --
336 :
338 rem elimina l'ultimo caratt. da l
    $
340 l$=left$(l$,len(l$)-1):printdl$;:
    return
342 :
344 rem --------- correzione --------
    --
346 :
348 if je=1 and lx=0 then fr=1:l$(0)=
    l$(0)+":rem[spc]ceksum[spc]errato
    !":goto452
350 printgg$;:lk=lb:ifje=1andj=0andcc
    =1thenlk=lb-1:goto354
352 ifje=1thena=j-1:gosub532
354 gosub552
356 for c=0 to lk:printl$(c):next
358 gosub552:printgg$;
360 jc=0:b=bz-1:dd=dl-1:s=12
362 z=0:d=0:printspc(s-1);:poke647,7
364 :
366 poke251,2
368 sys49152:a=peek(253):a$=chr$(a)
370 onpeek(254)goto 436,376,412,424
372 :
374 rem return
376 print:if jc<>lk then402
378 poke53265,11:rem fast
380 a=14:if je=1 thena=15
382 ifbb=10thenck=val(mid$(l$(jc),a))
    :goto386
384 a$=mid$(l$(jc),a):gosub564:ck=a
386 n=0:fori=0tojc-1
388 yy=len(l$(i))-11:yy=int(yy/(bz+1)
    ):fory=0to yy
390 ifbb=10thena=val(mid$(l$(i),12+4*
    y,3)):goto394
392 a$=mid$(l$(i),12+3*y,2):gosub564
394 n=n+a:nexty,i
396 poke53265,27:rem slow
398 ifck<>nthensysber:goto358
400 sysbok:goto452
402 : jc=jc+1:ifjc<lkthensysbok:goto3
    62
404 : dd=0:b=len(l$(jc))-14:s=14:a$=
    mid$(l$(jc),13,1):ifa$="*"thens=1
    5:b=b-1
406 : sysbok:goto362
408 :
410 rem crsr left
412 if(pos(0)=len(l$(jc))-1)or(z=b
    and d=dd)thensysber:goto368
414 printa$;:sysbip
416 z=z+1:ifz=b+1 thenz=0:d=d+1:print
    le$;
418 goto368
420 :
422 rem crsr right
424 ifz=0andd=0thensysber:goto368
426 z=z-1:ifz<0thenz=b:d=d-1:printri$
    ;
428 printri$;:sysbip
430 goto368
432 :
434 rem carattere
436 printa$;:sysbip
438 l$(jc)=left$(l$(jc),s+(b+2)*d+z-1
    )+a$+mid$(l$(jc),s+(b+2)*d+z+1)
440 if(z=b and d=dd)or(len(l$(jc))=
    pos(0))thenprintri$;:goto444
442 z=z+1:ifz=b+1thenz=0:d=d+1:printl
    e$;
444 goto368
```

```
446 :
448 rem -- append lines to seq file -
    -
450 :
452 am=0:ifccthenam=lb-(1andje=1andj=
    0)
454 gosub552:aj=0
456 :
458 printrs$1$(aj)
460 print#2,chr$(34)1$(aj)chr$(34):
    gosub542
462 aj=aj+1:if aj =< am then458
463 :
464 rem ----- esce o ritorna -------
465 :
466 printvv$;:poke647,5
468 if je=0 and cc=1 then192
470 if je=0 and cc=0 then194
472 if fr=1 then484
474 if fr=0 thenifje=1 and cc=1 then4
    86
476 if cc=0 and je=1 then284
478 :
480 rem ---------  termine  ---------
482 :
484 printgg$chr$(17)rv$"[10 spc]E
    [spc]R[spc]R[spc]O[spc]R[spc]E
    [10 spc]"
486 printvv$
488 poke198,0:close2:gosub542
490 f$=left$(fi$,16)+",s,r"
492 open2,8,2,f$:poke2,st:gosub542
494 :
496 poke152,2:ifpeek(2)>0then506
498 input#2,a$
500 poke2,st:print"[clr]"a$:print"got
    o496"
502 poke198,3
504 poke631,19:poke632,13:poke633,13:
    end
506 close2:close1:print"[clr]Un[spc]m
    omento..."
508 a=peek(43)+peek(44)*256
510 l=peek(a+0)+peek(a+1)*256:rem lin
    k
512 n=peek(a+2)+peek(a+3)*256:rem n.l
    in.
514 if n< 716 then a=1:print".";:goto
    510
516 poke43,peek(l):poke44,peek(l+1):
    list:end
518 :
520 rem attenzione: accertarsi che ne
    lla
522 rem linea 514 si confronti la var
    iabile 'n'
524 rem con il penultimo numero di li
    nea del programma
526 :
528 rem ------ definisce uq$ -------
530 :
532 uq$="":a1=(a*(bz+1)+11)/a%:a2=int
    (a1):ifa1>a2thena2=a2+1
534 lm=int(13/a2):fori=1toa2:uq$=uq$+
    up$:next:return
536 :
538 rem ------ errore disco --------
540 :
542 input#1,e1,e1$,e2,e3:ife1<20ore1=
    63thenreturn
544 print"[ctrl<2>][down]Disk[spc]err
    or:"e1;e1$;e2;e3:close2:close1:
    end
546 :
548 rem --- posiziona in prima riga -
    --
550 :
552 printup$uq$;:c=1
554 ifc<lkthenprintur$;:c=c+1:goto554
556 return
558 :
560 rem ----- converte dec in esa ---
    --
562 :
564 a=0:forc=1tolen(a$):t=fnde(asc(
```



```
    mid$(a$,c,1))):a=a*16+t:next:
    return
566 :
568 :
570 rem ----- lettura routine lm ----
    --
571 :
572 m=49152 :rem locazione inizio mem
    oria
574 fd=01    :rem flag (fd=00 data in
    esa, fd=01 data in dec.)
576 goto594
578 :
580 l=peek(66)*256+peek(65):return
582 print"errore[spc]nelle[spc]linee"
    l1"-"l-1:end
584 print"errore[spc]nella[spc]linea"
    l:end
586 reada$
588 iffdthena=val(a$):return
590 a=0:forx=1tolen(a$):t=fnde(asc(
    mid$(a$,x,1))):a=a*16+t:next:
    return
592 :
594 reada$:ifa$="**"then 606
596 : ifflag=0thengosub580:flag=1
598 : ifa$="*"thenflag=0:print:gosub5
    86:ifck<>athenl1=l:gosub580:goto5
    82
600 : ifflag=0thenck=0:goto594
602 : gosub 588:ifa>255thengosub580:
    goto584
604 pokem,a:m=m+1:ck=ck+a:print".";:
    goto594
606 gosub 586:if ck<>a then 582
608 :
610 return
612 :
614 data 169,000,133,204,133,198,165
616 data 198,240,252,173,119,002,166
618 data 252,240,011,201,065,048,002
620 data 201,071,016,003,076,116,192
622 data 201,072,208,004,169,048,208
624 data 080,201,077,208,004,169,049
625 data *,5349
626 data 208,072,201,044,208,004,169
628 data 050,208,064,201,046,208,004
630 data 169,051,208,056,201,074,208
632 data 004,169,052,208,048,201,075
634 data 208,004,169,053,208,040,201
636 data 076,208,004,169,054,208,032
637 data *,5045
638 data 201,085,208,004,169,055,208
640 data 024,201,073,208,004,169,056
642 data 208,016,201,079,208,004,169
644 data 057,208,008,201,048,048,021
646 data 201,058,016,017,162,001,208
648 data 010,162,002,208,006,162,003
649 data *,4359
650 data 208,002,162,004,076,197,192
652 data 166,251,208,015,201,042,240
654 data 235,201,020,240,235,201,174
656 data 240,235,076,000,192,166,251
658 data 224,001,208,015,201,042,240
660 data 222,201,020,240,214,201,013
661 data *,6472
662 data 240,206,076,000,192,166,251
664 data 224,002,240,003,076,000,192
666 data 201,013,240,190,201,029,240
668 data 190,201,157,240,190,076,000
670 data 192,133,253,134,254,162,000
672 data 142,119,002,120,169,020,133
673 data *,5869
674 data 205,164,211,070,207,174,135
676 data 002,177,209,176,017,230,207
678 data 133,206,032,036,234,177,243
680 data 141,135,002,174,134,002,165
682 data 206,041,127,032,028,234,169
684 data 001,133,204,088,096,169,015
685 data *,5541
686 data 162,045,160,165,032,063,193
688 data 169,033,162,006,160,005,032
690 data 073,193,162,255,076,083,193
692 data 169,015,162,000,160,247,032
694 data 063,193,169,017,162,040,160
```



```
 696 data 000,032,073,193,162,100,076
 697 data *,4647
 698 data 083,193,169,015,162,050,160
 700 data 247,032,063,193,169,017,162
 702 data 200,160,000,032,073,193,162
 704 data 050,076,083,193,141,024,212
 706 data 142,005,212,140,006,212,096
 708 data 141,004,212,142,001,212,140
 709 data *,4979
 710 data 000,212,096,160,255,136,208
 712 data 253,202,208,248,169,016,141
 714 data 004,212,169,000,141,001,212
 716 data 141,000,212,096
 717 data **,3492
```

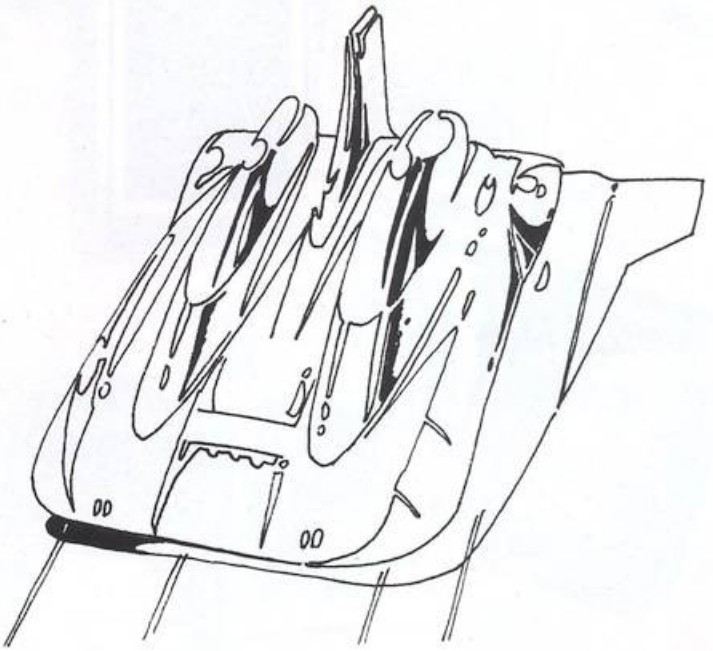
c64.datastore

```
  10 m=49152 :rem locazione inizio mem
     oria
  15 fd=00    :rem flag (fd=00 data in
     esa, fd=01 data in dec.)
  20 deffn de(x)=x-48+(x>57)*7:goto65
  25 :
  30 l=peek(66)*256+peek(65):return
  35 print"errore[spc]nelle[spc]linee"
     l1"-"l-1:end
  40 print"errore[spc]nella[spc]linea"
     l:end
  45 reada$
  50 iffdthena=val(a$):return
  55 a=0:forx=1tolen(a$):t=fnde(asc(
     mid$(a$,x,1))):a=a*16+t:next:
     return
  60 :
  65 reada$:ifa$="**"then 95
  70 : ifflag=0thengosub30:flag=1
  75 : ifa$="*"thenflag=0:print:gosub4
     5:ifck<>athenl1=l:gosub30:goto35
  80 : ifflag=0thenck=0:goto65
  85 : gosub 50:ifa>255thengosub30:
     goto40
  90 pokem,a:m=m+1:ck=ck+a:print".";:
     goto65
  95 gosub 45:if ck<>a then 35
 100 :
```

64mlx.datamaker

```
   0 data 0,0, 0, 0,0,0,0,0,0,0
   1 goto15
   2 h$="":rem trasforma in base esade
     cimale
   3 deffnesa(x)=(xand15)+48-((xand15)
     >9)*7
   4 h$=chr$(fnesa(t-int(t/16)*16))+h$
     :t=int(t/16):ift>0then4
   5 return
   6 iffd=0then h=ck:gosub2:print
     right$("000"+h$,4):return
   7 printmid$(str$(ck),2):return
   8 ln=ln+in
   9 v$=",":print"[ctrl<1>]0dA"mid$(
     str$(l1),2)v$mid$(str$(ln),2)v$;
  10 printstr$(l1)v$str$(lb)v$mid$(
     str$(fd),2)v$;
  11 printmid$(str$(ck),2)v$mid$(str$(
     m1),2)v$mid$(str$(m2),2)v$;
  12 printmid$(str$(dl),2)v$mid$(str$(
     in),2):return
  13 readll,ln,l1,lb,fd,ck,m1,m2,dl,in
     :return
  14 rem ---------------- mlx datamake
     r    by  a. de sabbata ----------
     --------
  15 ma=250:mb=65535
  16 poke53280,0:poke53281,0
  17 print"[2 home][clr][ctrl<8>]
     [ctrl<n>]":a=38:forj=1toa:x$=x$+"
     [spc]":next:e$="[up][spc]"+x$+"
     [2 up]"
  18 da$="[rvon]MLX.DATAMAKER[2 spc]by
     [spc]A.[spc]De[spc]Sabbata"
  19 forj=1to3:print"[spc][rvon]"x$:
     next:print"[home][2 down]"spc((a-
     len(da$))/2)da$g$"[down]"
  20 input"[down]Formato[spc]MLX[spc](
     s/n):[3 spc]n[3 right]";mx$:ifmx$
```

```
   <>"s"andmx$<>"n"thenprinte$:goto2
   0
21 mx=0:ifmx$="s"thenmx=1
22 input"[down]locaz.[spc]iniziale
   [2 spc]:[spc]";m1:ifm1<maorm1>mb-
   10thenprinte$:goto22
23 m1$=right$("[3 spc]"+str$(m1),5)
24 input"[down]locazione[spc]finale
   [spc]:[spc]";m2:ifm2=<m1orm2>mb
   thenprinte$:goto24
25 ifmxthenln=m1:goto52
26 input"[down]dec[spc]o[spc]esa
   [spc](d/e)[2 spc]:[3 spc]e
   [3 right]";b$:ifb$<>"d"andb$<>"e"
   thenprinte$:goto26
27 dm$="07":bz=3:ifb$="e"thendm$="10
   ":bz=2
28 print"[down]linea[spc]di[spc]iniz
   io[2 spc]:[3 spc]"m1$"[7 right]";
   :inputln
29 ifln<70thenprinte$:goto28
30 dm=int(70/(bz+1)):print"[down]dat
   a[spc]per[spc]linea[3 spc]:
   [3 spc]"dm$"[4 right]";:inputdl
31 ifdl<1ordl>dmthenprinte$:goto30
32 in$=str$(dl)
33 print"[down]incremento[7 spc]:
   [3 spc]"in$"[4 right]";:inputin
34 ifin<2or ln+(m2-m1)/dl*in>63999
   thenprinte$:goto33
35 input"[down]cksum[spc]totale[spc]
   o[spc]a[spc]blocchi[spc](t/b):
   [3 spc]b[3 right]";c$
36 ifc$<>"t"andc$<>"b"thenprinte$:
   goto35
37 lb=-1:ifc$="t"then41
38 input"[down]numero[spc]di[spc]lin
   ee[spc]per[spc]blocco[4 spc]:
   [3 spc]05[4 right]";lb
39 iflb<1orlb>20thenprinte$:goto38
40 :
41 rem -------------- costruzione li
   nee data formato cksum ----------
   ------
42 ii=0:ck=0:ll=ln:l1=-1:fd=0:ifb$="
   d"thenfd=1
43 l1=l1+1:print"[clr][ctrl<8>]";ln-
   ((in-1)andl1=lb)"data[spc]";:rem
   ** start **
44 ifl1=lbthenprint"*,";:t=ck:gosub6
   :l1=-1:ck=0:gosub9:print"goS13:g0
   43":goto64
45 a=peek(m1):iffdthena$=right$("00"
   +mid$(str$(a),2),3):printa$;:goto
   47
46 t=a:gosub2:printright$("0"+h$,2);
47 ii=ii+1:m1=m1+1:ck=ck+a
48 ifm1>m2thenprint:println+1"data
   [spc]**,";:t=ck:gosub6:goto61
49 ifii<dlthenprint",";:goto45
50 print:gosub8:print"goS13:g043":
   goto65
51 :
52 rem -------------- costruzione li
   nee data formato mlx ------------
   ------
53 ii=0:ck=0:ll=ln:dl=6:in=6:n=0:dim
   v(5)
54 print"[clr][ctrl<8>]"ln":";:v=ln-
   int(ln/256)*256:rem ** start **
55 a=peek(m1):a$=right$("00"+mid$(
   str$(a),2),3):printa$",";
56 forj=1to3:b=asc(mid$(a$,j,1)):n=n
   *10+b-48:next:v(ii)=n:n=0
57 ii=ii+1:m1=m1+1:ifii<dl then55
58 ck=v:forj=0to5:ck=(ck+v(j))and255
   :next:printright$("00"+mid$(str$(
   ck),2),3)
59 ifm1=<m2thengosub8:print"goS13:g0
   54":goto65
60 print
61 a=peek(43)+peek(44)*256
62 l=peek(a+0)+peek(a+1)*256
63 n=peek(a+2)+peek(a+3)*256:ifn<64
   thena=1:goto62
64 print"poke43,"peek(l)":poke44,"
   peek(l+1)":list"
65 poke198,4:poke631,19:poke632,13:
   poke633,13:poke634,13:end
```



GRUPPO EDITORIALE JACKSON

EDUCAZIONE TEC

LIBRI DI TESTO PER

Pietro Adorni
ELETTROTECNICA GENERALE
NIE 681455 N
pp. 412 • L. 24.000
Gli argomenti trattati sono quelli essenziali per una completa preparazione di base di elettrotecnica. Nessun aspetto teorico di definizione o di dimostrazione viene enunciato senza essere adeguatamente motivato dal punto di vista logico-funzionale. Gli esempi, ampiamente discussi, consentono di passare dal teorico al pratico con disinvoltura.

Paul B. Zbar
Joseph G. Sloop
DALL'ELETTROTECNICA ALL'ELETTRONICA INTEGRATA
Manuale di laboratorio
NIE 681469 Q
pp. 760 • L. 45.000
In questo testo vengono affrontate le tematiche riguardanti i corsi di esercitazioni pratiche di elettrotecnica, elettronica di base ed elettronica integrata. Ogni esperimento prevede l'indicazione degli obiettivi didattici da raggiungere, una introduzione tecnica, un sommario con un test di autovalutazione.

Mario Malcangi
SISTEMI, MODELLI E PROCESSI
Corso di sistemi d'automazione Vol. I
NIE 681451 J
pp. 200 • L. 18.000
Il libro fornisce una metodologia sistemistica orientata alle applicazioni nel campo dell'automazione, del controllo, dell'automatica in generale. Vengono presentati gli elementi tecnologici che portano alla realizzazione dei sistemi e gli strumenti di natura teorica e metodologica per l'analisi e la sintesi di sistemi di qualsiasi natura automatica.

Herbert Taub
Donald Schilling
FONDAMENTI DI ELETTRONICA INTEGRATA DIGITALE
NIE 681110 J
pp. 308 • L. 24.000
Dal famoso testo dei prof. Taub e Schilling, ampiamente utilizzato nelle università italiane, si è effettuata una riedizione adatta alle scuole medie superiori. La trattazione privilegia un approccio funzionale e conduce alla realizzazione sperimentale.

Dino Pellizzaro
MISURE ELETTRICHE
NIE 681447 Q
pp. 400 • L. 25.000
Vengono affrontate tutte le tematiche relative ad un corso di misure elettriche e laboratorio, con particolare risalto alle esercitazioni il cui scopo è spesso quello di ricavare sperimentalmente le leggi che governano l'elettrotecnica. La presentazione di data sheet, apparecchiature e strumenti, consente di affrontare gli aspetti relativi alla tecnologia e alle costruzioni.

Mauro Gargantini
Armando Zecchi
ELETTRONICA INTEGRATA LINEARE
NIE 681416 X
pp. 392 • L. 23.000
Un efficace manuale per poter impostare la progettazione circuitale sulle nuove tecnologie elettroniche e la loro realizzazione pratica. Viene approfondito lo studio dell'amplificatore operazionale e delle sue applicazioni. Viene affrontata tutta l'area di progettazione della microelettronica lineare.

Mario Malcangi
SISTEMI DIGITALI PER L'AUTOMAZIONE
Corso di sistemi d'automazione Vol. II
NIE 681453 L
pp. 200 • L. 18.000
Nella parte tecnologica vengono trattati gli elementi di base dei sistemi a microprocessore, delle unità programmabili di natura centrale e periferica, degli strumenti di comunicazione tra sistemi programmabili. Nella parte metodologica vengono discussi gli elementi fondamentali per l'analisi e lo sviluppo di sistemi programmabili.

Eugenio Piana
Pierfranco Ravotto
PROGETTARE CON L'ELETTRONICA DIGITALE
Dalla logica cablata al programmabile
NIE 681459 R
pp. 640 • L. 32.000
Sintetiche schede di "teoria" accompagnano 66 "esercitazioni" tutte rivolte alla comprensione ed all'uso di componenti in commercio di cui sono forniti i data sheet. È possibile realizzare in proprio, a scuola o a casa, la scheda necessaria per alimentare, montare e provare i diversi circuiti.

Thomas L. Floyd
CIRCUITI ELETTRICI
Corso di elettrotecnica generale
NIE 681471 A
pp. 672 • L. 35.000
Questo libro tratta gli argomenti essenziali relativi ai circuiti elettrici in corrente continua e in alternata, con particolare riguardo alle applicazioni e alla risoluzione dei problemi proposti a due livelli, uno relativamente basso, l'altro più impegnativo e stimolante. Alla fine del libro sono fornite le soluzioni.

Giuseppe Giuliano
MICROPROCESSORI
Architettura e programmazione
NIE 681461 X
pp. 252 • L. 20.000
Partendo dai concetti di base e dagli aspetti circuitali e di programmazione associati al microprocessore, la trattazione si concretizza nella descrizione di tre diverse MPU. Di esse una è del tutto generica e viene trattata quale base teorica per facilitare la comprensione del componente microprocessore. Le restanti rappresentano l'MPU Z80 e l'8086.

Mario Malcangi
SISTEMI, AUTOMAZIONE E CONTROLLO
Corso di sistemi d'automazione Vol. III
NIE 681393 B
pp. 192 • L. 18.000
Vengono affrontate le tecniche dell'acquisizione dati, il controllo, il trattamento numerico dei segnali e la comunicazione dei dati. Il libro comprende una parte a carattere metodologico: hardware della logica programmata, software, applicazioni in tempo reale e modelli di controllo.

Ugo Sgubbi
Santi Farina
Alessandro Gava
TELEMATICA DI BASE
NIE 681381 C
pp. 192 • L. 18.000
Un testo efficace per fornire ai futuri periti in telecomunicazione le conoscenze di base richieste dal mercato del lavoro. Viene analizzato il mondo della telematica, nei suoi aspetti fondamentali: dispositivi standard e sistemi di comunicazione. Particolare accento viene posto sui modem banda base e fonici di cui vengono descritti il funzionamento ed il miglior uso.

Nuovi str per una scuola

IN VENDITA PRESSO: libreria fiduciaria Jackson

ADEGUATI AI PROGETTI AMBRA ED ERGON

NICA SUPERIORE

GLI ISTITUTI TECNICI

Paul B. Zbar
Joseph G. Sloop
LABORATORIO DI ELETTRONICA INTEGRATA
NIE 681405 X
pp. 246 • L. 18.000
Le caratteristiche dei circuiti integrati lineari e degli amplificatori operazionali. Vengono proposte esperienze per utilizzare circuiti integrati per la generazione dei segnali e i circuiti PLL. Si mostrano alcune utilizzazioni dei circuiti integrati digitali e dei convertitori A/D e D/A.

Renzo Traversini
MICROELETTRONICA: TECNOLOGIE E DISPOSITIVI
Corso di tecnologie elettroniche Vol. II
NIE 681126 W
pp. 192 • L. 18.000
Le tecnologie più utilizzate per la fabbricazione dei circuiti integrati al silicio, presentate in stretto legame con le strutture fisiche dei componenti (diodi, transistor bipolari, transistori MOS) che con esse si realizzano.

Mariangela Botti
DAL PROBLEMA AL PROGRAMMA
NIE 681352 J
pp. 328 • L. 24.000
Obiettivo è quello di fornire agli allievi che iniziano lo studio dell'informatica gli elementi fondamentali per la risoluzione di un problema sino alle soglie della sua codifica. I numerosi esercizi risolti e proposti giocano un ruolo di assoluto rilievo: attraverso l'analisi e la descrizione degli algoritmi vengono presentate le strutture fondamentali della programmazione, gli array e le subroutine.

Felice Tarantini
COMMUTAZIONE TELEFONICA AUTOMATICA
NIE 681403 Q
pp. 220 • L. 23.000
Fornisce agli studenti la conoscenza delle tecniche di commutazione automatica affermatesi in Italia a partire dai primi sistemi automatici fino agli attuali sistemi interamente elettronici, evidenziando l'aspetto funzionale tecnologico degli autocommutatori. Vengono poi descritti il funzionamento della rete telefonica nazionale e le tecniche di commutazione nelle centrali.

Paul B. Zbar
Joseph G. Sloop
LABORATORIO DI ELETTROTECNICA
NIE 681399 M
pp. 302 • L. 21.000
L'uso del multimetro, la realizzazione di alcune reti elettriche, le caratteristiche di alcuni campi elettrici e magnetici, l'utilizzo dell'oscilloscopio, alcune esercitazioni sui condensatori, sugli induttori e quindi sui circuiti RL, RC, RLC serie e parallelo.

Fosco Bellomo
ELEMENTI PASSIVI TECNOLOGIE E DISPOSITIVI
Corso di tecnologie elettroniche Vol. I
NIE 681457 P
pp. 352 • L. 24.000
Vengono introdotti i fondamenti tecnologici relativi ai materiali utilizzati nel campo elettronico e ai parametri meccanici, fisici e chimici che ne determinano la scelta. Vengono analizzate le tecnologie costruttive degli elementi cosiddetti passivi, quali resistori, condensatori, induttori, legandole alle strutture fisiche dei componenti e agli aspetti applicativi degli stessi.

Peter Bishop
INFORMATICA GENERALE
NIE 681473 J
pp. 540 • L. 24.000
Cod.: SD668
Tutti gli aspetti teorici e pratici della materia informatica, come previsto dai programmi ministeriali per gli Istituti Tecnici Industriali e Commerciali. Si articola in cinque sezioni: i principi dell'elaborazione dell'informazione, la struttura dell'elaboratore e l'architettura dei sistemi, il software di sistema, l'organizzazione dei dati e le applicazioni.

Giuseppe Saccardi
TELEMATICA DAI PROTOCOLLI ALLE RETI
NIE 681449 X
pp. 240 • L. 24.000
Il mondo della telematica partendo dall'evoluzione verso le reti telematiche, descrivendo i protocolli di trasmissione sincroni oggi più usati, cioè i BSC, l'SDLC, l'HDLC. Sono descritti i dispositivi per reti telematiche, di multiplazione e di concentrazione, il che permette di comprendere il passaggio da una rete convenzionale ad una commutazione di pacchetto.

Paul B. Zbar
Joseph G. Sloop
LABORATORIO DI ELETTRONICA DI BASE
NIE 681401 W
pp. 272 • L. 18.000
Il funzionamento dei multimetri e degli oscilloscopi per la misura delle grandezze elettriche fondamentali, esercitazioni sui diodi a semiconduttori sui circuiti elettronici che li utilizzano, esperimenti sugli alimentatori, sui transistori a funzione e ad effetto di campo.

Fosco Bellomo
MICROELETTRONICA NUOVE TECNOLOGIE
Corso di tecnologie elettroniche Vol. III
NIE 681467 W
pp. 200 • L. 18.000
L'obiettivo in questo testo è quello di fornire, a completamento dei programmi ministeriali del triennio degli Istituti Tecnici, una conoscenza approfondita sulle nuove tecnologie, con la descrizione delle mete raggiunte e dei vantaggi ottenuti ed ipotizzando, in alcuni casi, quali potranno essere i successivi sviluppi.

Salvatore Consentino
ORGANIZZAZIONE INDUSTRIALE STUDI DI FABBRICAZIONE E DISEGNO
NIE 681463 K
pp. 216 • L. 22.000
Si articola in tre parti: la prima mette in rilievo l'importanza della grafica e delle tecniche di lavoro legate al CAD. Una seconda parte è dedicata alla struttura dell'impresa industriale nelle sue principali funzioni e sono descritti limiti e vantaggi dei modelli nella ricerca operativa. L'ultima sezione riguarda gli aspetti relativi alla produzione.

Paul H. Young
COMUNICAZIONI ELETTRICHE
Corso di radioelettronica
NIE 681465 M
pp. 498 • L. 34.000
Partendo dalla descrizione degli amplificatori a radiofrequenza, degli oscillatori, degli spettri dei segnali e dei sistemi di modulazione d'ampiezza, si passa alla trattazione dei più moderni circuiti di trasmissione e di ricezione. Ampio spazio viene dedicato alla comunicazione digitale e alle tecniche di trasmissione dati.

umenti che cambia

DISTRIBUZIONE ESCLUSIVA:

La Nuova Italia

Un comodo Merge

Concateniamo i nostri programmi eliminando la fase del List altrimenti inevitabile

Molte tecniche di Merge sono state illustrate nell'immensa biblioteca dedicata al C-64, ma quella che vi faremo conoscere tra poco elimina un fastidioso procedimento altrimenti indispensabile nell'uso di questo atipico comando BASIC.

Sicuramente molti di voi si saranno trovati almeno una volta obbligati a dover effettuare un merge di due o più programmi Basic. Questa operazione diventa particolarmente utile quando si vuole arricchire un nuovo programma di subroutine già precedentemente sviluppate. Infatti, dopo tanti anni di programmazione sul 64 molti utenti si sono creati delle librerie esclusivamente riservate alle varie subroutine che nei requisiti di ogni programma degno di tale nome si rendono indispensabili. Il programma pubblicato a seguito di questo articolo, vi permetterà di unire un programma già in memoria con uno residente su disco. In tal modo vi eviterete noiose ricopiature di routine e, soprattutto, ridurrete a zero gli errori che normalmente si verificano durante la copiatura dei programmi. Il programmatore ne ricava un guadagno nei termini di tempo e soprattutto nell'equilibrio psico-fisico, risparmiandosi inutili arrabbiature.

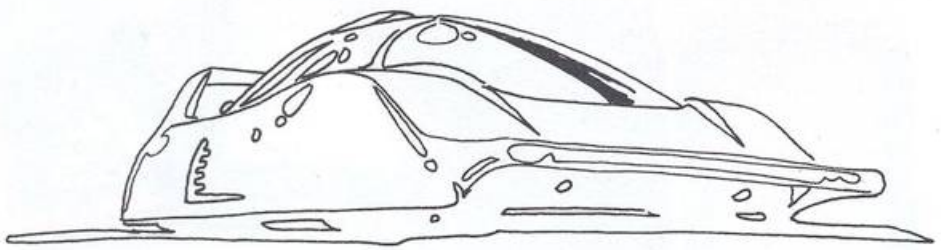
Per semplificare la fase di scrittura del programma, esso è stato trasformato in semplici righe DATA che vengono normalmente pokate in memoria dalla parte Basic del programma.

Il codice macchina del programma Merge, questo il suo nome, risiede nella parte vicina alla Top Memory. Siccome il Sistema Operativo riserva questa zona della memoria all'immagazzinamento delle variabili, abbiamo dovuto proteggere il programma Merge da eventuali intrusioni dei vettori. A questo punto ci è venuto incontro amichevolmente il S. O.; infatti, cambiando solamente il valore di due locazioni nella memoria, si possono abbassare i puntatori alla Top Memory addomesticandoli a nostro piacimento. Dopo aver agito su queste due locazioni il C-64 prenderà i valori in esse indicati come il tetto della memoria e ignorerà completamente qualsiasi informazione si trovi oltre tale indirizzo. Se date un'occhiata al listato di Merge noterete le due poke che si riferiscono alle locazioni 55 e 56. La locazione 56 contiene il Byte Alto dell'indirizzo del tetto della memoria mentre la locazione 55 ne contiene il Byte Basso. Non allarmetevi, non è così complicato come può sembrare. Moltiplicando il 'peek' della locazione 56 (Byte Alto) per 256 e aggiungendovi il valore della locazione 55 (Byte Basso) otterremo la locazione del tetto della memoria.

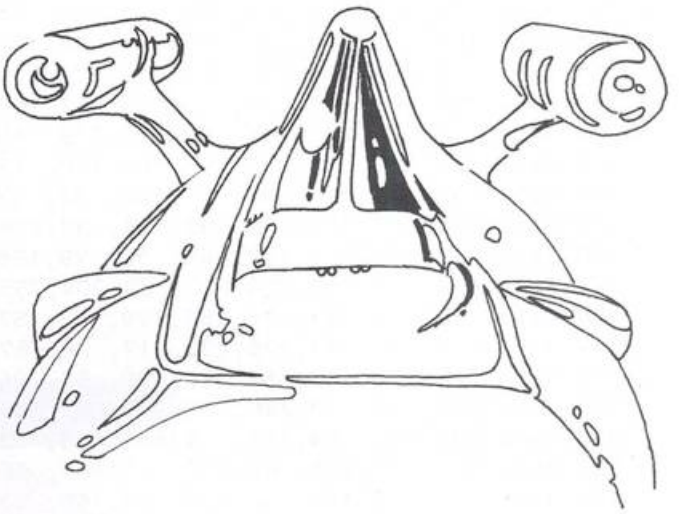
Ora, ogniqualvolta noi riduciamo il valore nella locazione 56 di una unita' abbasseremo il Top Memory di 256 byte (ricordate: 56 contiene il valore del Byte Alto), ovviamente sottraendo una unità al valore contenuto nella locazione 55 abbasseremo il tetto della memoria di un solo byte (ricordate: 55 contiene il valore del Byte Basso).

Lo statement dedicato all'abbassamento del tetto della memoria dovrebbe sempre essere il PRIMO statement nei vostri programmi e le due Poke devono essere seguite dal comando 'CLR', in quanto esso istruisce il sistema operativo in modo da portare altri puntatori di memoria nella riga dove risiedono i puntatori al Top Memory.

Dopo questa breve parentesi esplicativa sull'uso dell'abbassamento del tetto della memoria, ritorniamo al programma Merge. Mentre i valori DATA vengono pokati in memoria, non visibile ma sicuramente funzionante, un meccanismo di addizioni tiene aggiornato un contatore che, sommando i vari valori, alla fine ne confronta la somma con il numero contenuto nella riga 140. Se i due numeri sono differenti vorra' dire che avete sbagliato qualcosa nella copiatura del listato, altrimenti comparirà un messaggio con delle istruzioni sull'uso di Merge.



```
100 rem merge programmi basic per c-64
110 poke 56,125:poke 55,0:clr
120 for j=32000 to 32467:read x
130 poke j,x:ch=ch+x:next
140 if ch<>49385 thenprint"
    "Errore nei data!":end
150 print"      Merge programmi
    Basic per C-64"
155 print
160 print" Caricare il primo programma
    in RAM, scrivere SYS 32000 e ";
170 print"seguire le istruzioni."
175 new:rem salvare prima dell'esecuzione
180 data 169,  0,133,183,169,147, 32,210
190 data 255,162,  0, 32, 79,126, 32,228
200 data 255,201, 83,240,  7,201, 78,208
210 data 245, 76,134,227,162, 73, 32, 79
220 data 126, 32,207,255,201, 20,240, 18
230 data 201, 13,240, 14,166,183,224, 16
240 data 240,231,157, 60,  3,232,134,183
250 data 208,231,166,183,240,219,169, 44
260 data 157, 60,  3,232,169, 80,157, 60
270 data   3,232,134,183,169, 13, 32,210
280 data 255,169, 13,133,184,133,185,169
290 data   8,133,186,169,  3,133,188,169
300 data  60,133,187, 32,192,255,162, 13
310 data  32,198,255, 32, 57,126, 32, 57
320 data 126, 32,228,255,208, 24, 32,228
330 data 255,208, 22,162, 94, 32, 79,126
340 data 169, 13, 32,195,255, 32,204,255
350 data  32, 89,166, 76,134,227, 32, 57
360 data 126, 32, 57,126,133, 20, 32, 57
370 data 126,133, 21,160,  0, 32, 57,126
380 data 153,  0,  2,240,  3,200,208,245
390 data 200,152, 24,105,  4,133, 11, 32
400 data  19,166,144, 68,160,  1,177, 95
410 data 133, 35,165, 45,133, 34,165, 96
420 data 133, 37,165, 95,136,241, 95, 24
430 data 101, 45,133, 45,133, 36,165, 46
440 data 105,255,133, 46,229, 96,170, 56
450 data 165, 95,229, 45,168,176,  3,232
460 data 198, 37, 24,101, 34,144,  3,198
470 data  35, 24,177, 34,145, 36,200,208
480 data 249,230, 35,230, 37,202,208,242
490 data  32, 99,166, 32, 51,165, 24,165
500 data  45,133, 90,101, 11,133, 88,164
510 data  46,132, 91,144,  1,200,132, 89
520 data  32,184,163,165, 20,164, 21,141
530 data 254,  1,140,255,  1,165, 49,164
540 data  50,133, 45,132, 46,164, 11,136
550 data 185,252,  1,145, 95,136, 16,248
560 data  32, 99,166, 32, 51,165, 76,113
570 data 125, 32,228,255,166,144,240, 13
580 data  32,204,255,162,105, 32, 79,126
590 data 104,104, 76,128,125,170, 96,189
600 data  91,126,240,  6, 32,210,255,232
610 data 208,245, 96, 13, 13, 18, 32, 32
```



```
620 data  77, 69, 82, 71, 69, 32, 80, 82
630 data  79, 71, 82, 65, 77, 77, 73, 32
640 data  45, 32, 67, 79, 77, 77, 46, 80
650 data  82, 79, 70, 46, 45, 32, 56, 51
660 data  48, 53, 50, 50,146, 13, 13, 80
670 data  82, 73, 77, 79, 32, 80, 82, 79
680 data  71, 46, 32, 67, 65, 82, 73, 67
690 data  65, 84, 79, 32, 63, 32, 40, 83
700 data  47, 78, 41,  0, 13, 13, 78, 79
710 data  77, 69, 32, 68, 69, 76, 32, 83
720 data  69, 67, 79, 78, 68, 79, 63, 32
730 data   0, 13, 13, 77, 69, 82, 71, 69
740 data  32, 79, 75,  0, 13, 13, 77, 69
750 data  82, 71, 69, 32, 78, 69, 71, 65
760 data  84, 79, 32,  0
```

```
100 rem    merge programmi basic 4.0
110 poke 53,125:poke 52,0:clr
120 for j=32000 to 32476:read x
130 poke j,x:ch=ch+x:next
140 if ck<>51236 thenprint"Errore
    nei data!":end
150 print"    Merge per programmi in
    Basic 4.0"
155 print
160 print" Caricare il primo programma
    in RAM, scrivere SYS 32000 e ";
170 print"seguire le istruzioni."
175 new:rem salvare prima dell'esecuzione
180 data 169,  0,133,209,169,147, 32,210
190 data 255,162,  0, 32, 79,126, 32,228
200 data 255,201, 83,240,  7,201, 78,208
210 data 245, 76,255,179,162, 73, 32, 79
220 data 126, 32,207,255,201, 20,240, 18
230 data 201, 13,240, 14,166,209,224, 16
```

```
240 data 240,231,157,131,  2,232,134,209
250 data 208,231,166,209,240,219,169, 44
260 data 157,131,  2,232,169, 80,157,131
270 data   2,232,134,209,169, 13, 32,210
280 data 255,169, 13,133,210,133,211,169
290 data   8,133,212,169,  2,133,219,169
300 data 131,133,218, 32, 99,245,162, 13
310 data  32,175,247, 32, 57,126, 32, 57
320 data 126, 32,228,255,208, 24, 32,228
330 data 255,208, 22,162,103, 32, 79,126
340 data 169, 13, 32,226,242, 32,204,255
350 data  32,233,181, 76,255,179, 32, 57
360 data 126, 32, 57,126,133, 17, 32, 57
370 data 126,133, 18,160,  0, 32, 57,126
380 data 153,  0,  2,240,  3,200,208,245
390 data 200,152, 24,105,  4,133,  5, 32
400 data 163,181,144, 68,160,  1,177, 92
410 data 133, 32,165, 42,133, 31,165, 93
420 data 133, 34,165, 92,136,241, 92, 24
430 data 101, 42,133, 42,133, 33,165, 43
440 data 105,255,133, 43,229, 93,170, 56
450 data 165, 92,229, 42,168,176,  3,232
460 data 198, 34, 24,101, 31,144,  3,198
470 data  32, 24,177, 31,145, 33,200,208
480 data 249,230, 32,230, 34,202,208,242
490 data  32,255,181, 32,182,180, 24,165
500 data  42,133, 87,101,  5,133, 85,164
510 data  43,132, 88,144,  1,200,132, 86
520 data  32, 80,179,165, 17,164, 18,141
530 data 254,  1,140,255,  1,165, 46,164
540 data  47,133, 42,132, 43,164,  5,136
550 data 185,252,  1,145, 92,136, 16,248
560 data  32,255,181, 32,182,180, 76,113
570 data 125, 32,228,255,166,150,240, 13
580 data  32,204,255,162,114, 32, 79,126
590 data 104,104, 76,128,125,170, 96,189
600 data  91,126,240,  6, 32,210,255,232
610 data 208,245, 96, 13, 13, 18, 32, 32
620 data  77, 69, 82, 71, 69, 32, 80, 82
630 data  79, 71, 82, 65, 77, 77, 73, 32
640 data  45, 32, 67, 79, 77, 77, 46, 80
650 data  82, 79, 70, 46, 45, 32, 56, 51
660 data  48, 53, 50, 50,146, 13, 13, 80
670 data  82, 73, 77, 79, 32, 80, 82, 79
680 data  71, 46, 32, 67, 65, 82, 73, 67
690 data  65, 84, 79, 32, 63, 32, 40, 83
700 data  47, 78, 41,  0, 13, 13, 68, 82
710 data  73, 86, 69, 35, 32, 38, 32, 78
720 data  79, 77, 69, 32, 68, 69, 76, 32
730 data  83, 69, 67, 79, 78, 68, 79, 63
740 data  32,  0, 13, 13, 77, 69, 82, 71
750 data  69, 32, 79, 75,  0, 13, 13, 77
760 data  69, 82, 71, 69, 32, 78, 69, 71
770 data  65, 84, 79, 32,  0
```

ATTENZIONE! Nella riga 175 potete notare il comando 'NEW', ricordate quindi di salvare il programma prima di dare il RUN.

Per concatenare due programmi procedete come segue:

a) LOAD e RUN di programma Merge;

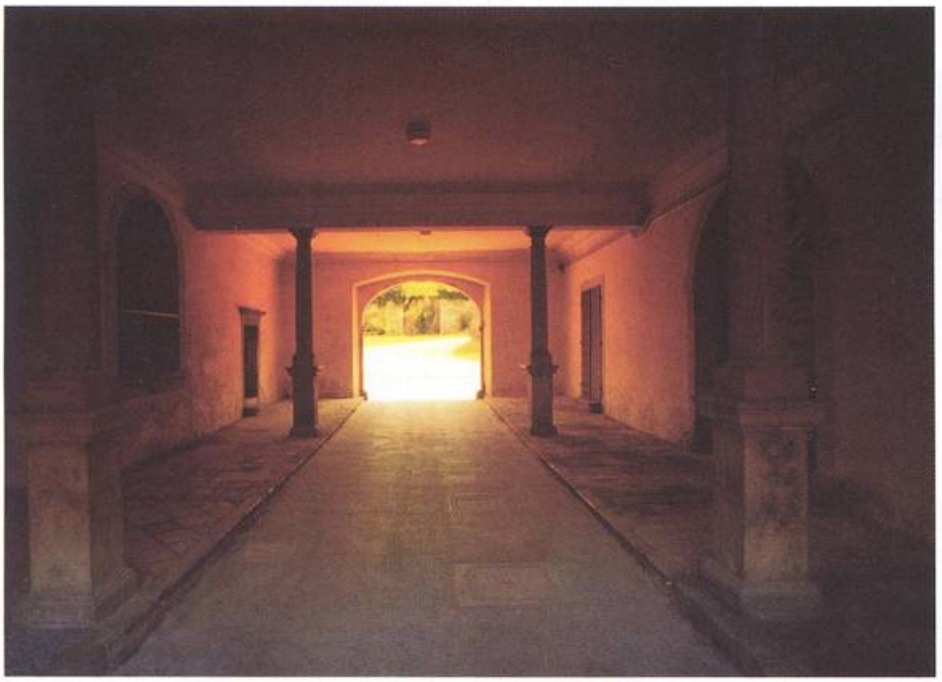
b) LOAD del primo programma Basic in memoria nel modo usuale;

c) scrivere SYS 32000 e premere il tasto RETURN;

d) scrivere S in risposta alla domanda 'primo programma caricato?';

e) scrivere il nome del file che volte concatenare al primo e premere RETURN.

f) a questo punto il merge viene effettuato.



Se tutto avviene regolarmente, alla fine compare il messaggio 'MERGE OK' e l'intero programma (prog. 1 + prog. 2) risiede nella memoria del computer, se volete aggiungere ulteriori programmi a questo dovete semplicemente ripetere le istruzioni dal punto b) al punto e) viste sopra.

Se si verifica un errore (file not found, read error eccetera) compare il messaggio 'MERGE NEGATO'.

Note

È molto importante tenere presente che eventuali numeri di linea uguali nei due programmi non possono coesistere, per cui il programma sovrappone la riga del secondo programma alla stessa riga del primo programma.

Sperando di avervi offerto un utile strumento di lavoro nella stesura dei vostri listati, lasciamo ora al vostro ingegno la creazione di utili programmi da inserire nelle pagine della bibbia Commodore, scusate, intendevamo dire nelle pagine di Commodore Professional.

NOVITÁ ASSOLUTA

PER IL NUOVO COMMODORE 64PC

CORSO DI MUSICA IN 15 LEZIONI

i primi 3 fascicoli
con cassette
e la tastiera
musicale, a sole

L.15.000

GRUPPO EDITORIALE JACKSON

IN EDICOLA SCEGLI IL MEGLIO SCEGLI JACKSON

IN EDICOLA

Libro mastro per il bowling

Durante la stagione del bowling, archiviate i vostri successi con questo database

È facile, durante i mesi freddi e umidi, ritrovarsi spesso a passare delle piacevoli serate con gli amici su qualche pista di bowling. Il gioco consiste nel lanciare tre palle (bowls) di legno per ogni giocatore contro una serie di nove o dieci birilli disposti, a forma di triangolo, al fondo di una pista ben levigata. Ogni serie di due palle costituisce un lancio, dieci lanci un gioco. Se questo passatempo impegna varie serate invernali, ci si accorge con evidente piacere che in un lasso di tempo molto breve ci si è trasformati in abilissimi giocatori di questo sport. Il migliorare i punteggi delle partite e lo stabilire serata dopo serata dei nuovi record comporterà anche un problema di come questi punteggi dovranno essere registrati e questo problema aumenterà la propria complessità qualora la gara per il miglior punteggio coinvolga più giocatori.

In tal caso vi sarà indubbiamente d'aiuto un programma che vi tenga aggiornati sui vari punteggi e record ottenuti. Questo 'Libro Mastro per il Bowling', e' stato scritto per il Commodore 128 in modalità 80 colonne ed è un foglio elettronico che è in grado di immagazzinare i punteggi delle partite. Può eseguire anche le seguenti operazioni: calcolare i punteggi totali e le medie per ciascuna sessione di bowling e mantenere aggiornati i punteggi e le medie da sessione a sessione. Creando un file per ogni singola persona o gruppo, sarà possibile utilizzare facilmente questo programma per registrare le medie individuali e i punteggi delle squadre in gara.

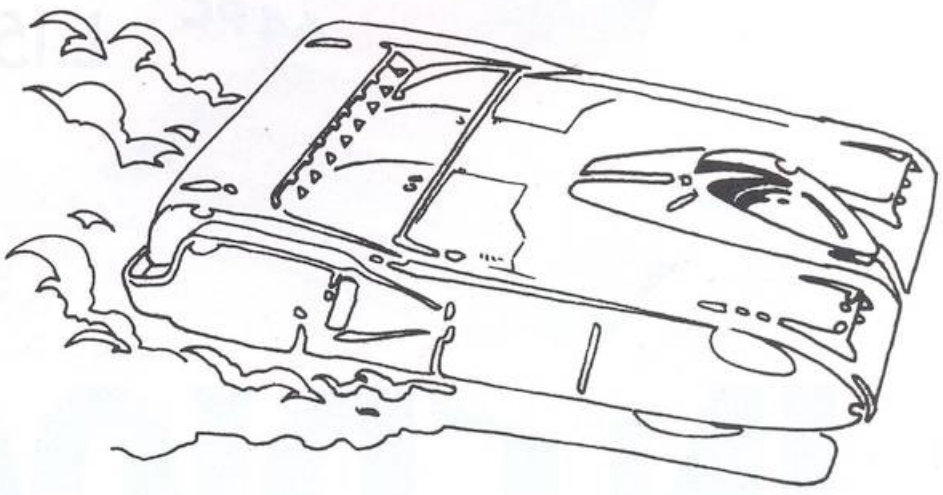
Il menù principale

Passiamo ora a spiegare più dettagliatamente il funzionamento del programma in questione e le sue possibili applicazioni. Quando si effettua il Run del programma vedrete comparire sullo schermo per prima cosa il menù principale elencante le sue sei diverse opzioni che prevedono: la possibilità di caricare in memoria un file; di registrare un file su dischetto; di esaminare le varie pagine dello schermo per potervi apportare delle eventuali modifiche, delle aggiunte o delle sostituzioni; di visualizzare la directory del disco; di stampare un file o, eventualmente, di uscire dal programma. Per eseguire una qualsiasi di queste scelte basterà seguire semplicemente il prompt corrispondente all'opzione desiderata. Premendo solamente il tasto Return ad ogni prompt si verrà rinviati al menù principale ed il programma non eseguirà la routine del prompt, in questo modo si può ovviare ad errori o ripensamenti dell'ultimo minuto.

Il programma crea dei file sequenziali e aggiunge ai loro nomi, in modo automatico, il suffisso '.tp' per distinguerli da altri file eventualmente presenti sul dischetto. Essendo automatica l'aggiunta di tale suffisso, non ci sara' bisogno di digitarlo durante le operazioni di Load o di Save che effettuerete su un file.

Input dei dati ed elaborazioni

Il programma prevede l'utilizzo di quattro pagine. Ogni pagina permette l'inserimento di 13 sessioni di partite di



```
bowling 80

    10 graphic5,1: printchr$(142): x=15:
       gosub 2070
    20 print"[6 down]"tab(28)chr$(2);"
       [cmdr<3>][3 spc]b[2 spc]o[2 spc]w
       [2 spc]l[2 spc]i[2 spc]n[2 spc]g
       [3 spc]"
    30 printchr$(x);"[3 down]"tab(34)"
       [ctrl<2>]Q[3 spc]Q[3 spc]Q[3 spc]
       Q"
    40 printchr$(x);tab(36)"Q[3 spc]Q
       [3 spc]Q"
    50 printchr$(x);tab(38)"Q[3 spc]Q"
    60 printchr$(x);tab(40)"Q"
    70 printchr$(8)chr$(14)
    80 dimd$(53),a(10,53),tb(10):gosub16
       40
    90 fori=1to10:readth(i):next
   100 data 2,9,17,25,33,42,49,56,67,73
   110 printchr$(147)
   120 :
   130 gosub1520
   140 getkeyk$
   150 k=val(k$):ifk<1ork>6then140
   160 printchr$(7)
   170 ifk=6thenprint"[2 home][clr]":end
   180 onkgosub210,320,430,980,1180
   190 goto140
   200 :
   210 window11,2,69,22:print"[down]"
       tab(12)chr$(15)"[ctrl<8>]Load
       [spc]File"
   220 window12,8,68,17:print"[down]
       [cmdr<7>][2 spc]Filename:[spc]
       [ctrl<2>]";:l=12:ty=4:gosub1690:
       iffl=0then290
   230 f$=v$+".tp"
   240 dopen#2,(f$),r
   250 print:print:print"[cmdr<3>]
       [2 spc]Stato[spc]disco:[spc]
       [ctrl<2>]"ds$:ifds<>0thengosub196
       0:goto280
   260 fori=1to52:input#2,d$(i):next
   270 fori=1to10:forj=1to52:input#2,a(i
       ,j):next:next
   280 dclose#2
   290 window12,8,68,17,1:pg=0:p=1:sc=0
   300 window11,2,69,22:print"[down]"
       tab(12)chr$(143)"[ctrl<6>]Load
       [spc]File":return
   310 :
   320 window11,2,69,22:print"[down]"
       tab(39)chr$(15)"[ctrl<8>]Save
       [spc]File"
   330 window12,8,68,17:print"[down]
       [cmdr<7>][2 spc]Filename:[spc]
       [ctrl<2>]";:l=12:ty=4:gosub1690:
       iffl=0then400
   340 f$=v$+".tp":scratch (f$)
   350 dopen#2,(f$),w
   360 print:print:print"[cmdr<3>]
       [2 spc]Stato[spc]Disco:[spc]
       [ctrl<2>]"ds$:ifds<>0thengosub196
       0:goto390
   370 fori=1to52:print#2,d$(i):next
   380 fori=1to10:forj=1to52:print#2,a(i
       ,j):next:next
   390 dclose#2
   400 window12,8,68,17,1:pg=0:p=1:sc=0
   410 window11,2,69,22:print"[down]"
       tab(39)chr$(143)"[ctrl<6>]Save
       [spc]File":return
   420 :
   430 gosub2080:if sc then print"
       [cmdr<7>]";: goto650
   440 sc=1:print"[2 home][clr][cmdr<3>]
       [rvon][2 spc]B[spc]o[spc]w[spc]l
       [spc]i[spc]n[spc]g:[64 spc]";
   450 print"[home]"tab(17)left$(f$,(len
       (f$)-3))
   460 print"[home]"tab(66)"[rvon]Pag.:"
       p
   470 print"[rvon][cmdr<1>][80 spc]";
   480 print"[rvon][ctrl<2>][2 spc]Dati
       [3 spc][shift<->][spc]Giochi
       [shift<->][spc]Punti[spc]
       [shift<->][spc]Punti[spc]
       [shift<->][spc]Punti[spc]
       [shift<->][spc]Totale[shift<->]
       [spc]Med.[cmdr<7>][shift<->][spc]
       Giochi[shift<->][2 spc]Totale
       [spc][shift<->][spc]Media"
   490 fori=1+pgto13+pg
   500 print"[ctrl<2>][rvon][spc][rvoff]
       [spc][ctrl<6>]";d$(i);:tb=10
   510 forj=2 to 6
   520 printtab(th);:printusing"#####";a
       (j,i);
   530 th=th+8:next
   540 printtab(51);:printusing"###";a(7
       ,i);
   550 printtab(58);:printusing"###";a(8
       ,i);
   560 printtab(66);:printusing"######";
       a(9,i);
   570 printtab(75);:printusing"###";a(1
       0,i);
   580 print"[spc][ctrl<2>][rvon][spc]
       [rvoff]";:next
   590 print"[rvon][ctrl<2>][32 spc]Somm
       ario[spc]Comandi[32 spc][ctrl<7>]
       "
   600 fori=1to6: print"[rvon][ctrl<2>]
       [spc][78 left][spc]";:next
   610 print"[rvon][ctrl<2>][80 spc]
       [ctrl<7>]"
   620 print"[2 home][17 down]":print
       tab(5)"[ctrl<8>][uparrow][spc]=
       [spc]Ingresso[spc]dati[14 spc]CON
       TROL[spc]1,2,3,4[spc]=[spc]Vedi
       [spc]Pagina[spc]#"
   630 printtab(5)"[backarrow][spc]=
```

```
    [spc]Calcolo[2 spc]Totali[12 spc]
    CONTROL[spc]F[spc]=[spc]Cerca
    [spc]serie[spc]giochi[spc]200/500
    "
640 printtab(5)"M[spc]=[spc]Return
    [spc]a[spc]Menu'[spc]Princip.
    [4 spc]CONTROL[spc]C[spc]=[spc]Cr
    ea[spc]nuovo[spc]file[cmdr<7>]"
650 gosub1910:remchar1,2,3,d$(r+pg),1
660 getkeyk$
670 ifk$="[left]"thenbegin:
680 gosub1870:c=c+1:ifc>5thenc=1:r=r+
    1:ifr>13thenr=13
690 gosub1910:bend
700 ifk$="[right]"thenbegin:
710 gosub1870:c=c-1:ifc<1thenc=1
720 gosub1910:bend
730 ifk$="[up]"thenbegin:
740 gosub1870:r=r-1:ifr<1thenr=1
750 gosub1910:bend
760 ifk$="[down]"thenbegin:
770 gosub1870:r=r+1:ifr>13thenr=13
780 gosub1910:bend
790 ifk$="m"thenprintchr$(7);:gosub20
    70:return
800 ifk$="[backarrow]"thenprintchr$(7
    );:gosub1250:rem totali
810 ifk$="[ctrl<2>]"thenpg=13:p=2:sc=
    0:printchr$(7);:goto430:rem pagin
    a 2
820 ifk$="[ctrl<1>]"thenpg=0:p=1:sc=0
    :printchr$(7);:goto430:rem pagina
     1
830 ifk$="[ctrl<3>]"thenpg=26:p=3:sc=
    0:printchr$(7);:goto430:rem pagin
    a 3
840 ifk$="[ctrl<4>]"thenpg=39:p=4:sc=
    0:printchr$(7);:goto430:rem pagin
    a 4
850 ifk$="[ctrl<f>]"thenprintchr$(7);
    :gosub1400:rem trovato
860 ifk$="[ctrl<c>]"thenprintchr$(7);
    :gosub2000:goto430:rem nuovo file
870 ifk$<>"[uparrow]"then660
880 printchr$(7);
890 print"[home][down][rvon][cmdr<1>]
    [2 spc]Enter:[spc]";
900 ifc=1thenl=5:ty=4:gosub1690:iffl
    thend$(r+pg)=v$
910 ifc=2thenl=2:ty=2:gosub1690:iffl
    thena(c,r+pg)=val(v$)
920 ifc>2thenl=4:ty=2:gosub1690:iffl
    thena(c,r+pg)=val(v$)
930 gosub 1870
940 print"[home][down][cmdr<1>][rvon]
    [18 spc]"
950 k$="[left]": goto670
960 goto660
970 :
980 window11,2,69,22:print"[2 down]"
    tab(39)chr$(15)"[ctrl<8>]Stampa
    [spc]File"
990 window12,8,68,17:print"[down]
    [cmdr<7>][2 spc]Stampante[spc]pro
    nta[spc][y/n]
1000 getkeyk$:ifk$<>"y"then1150
1010 print"[down][cmdr<1>][spc]Stampan
     do..."
1020 open4,4,0
1030 print#4,"[5 spc]file:[spc]";left$
     (f$,(len(f$)-3)):print#4
1040 print#4,"[5 spc]dati[7 spc]#
     [3 spc]punti[2 spc]punti[2 spc]pu
     nti[2 spc]punti[3 spc]media
     [4 spc]#[2 spc]totale[2 spc]med."
1050 print#4,"[5 spc]";
1060 fori=1to68:print#4,"[shift<*>]";:
     next:print#4
1070 i=1
1080 do until a(2,i)=0
1090 print#4,"[5 spc]";
1100 print#4,using "#####";d$(i);:
     print#4,"[2 spc]";
1110 forj=2to10
1120 print#4,using "#####";a(j,i);:
     print#4,"[2 spc]";
1130 next:print#4:i=i+1:loop
1140 print#4:close4
1150 window12,8,68,17,1:pg=0:p=1:sc=0
1160 window11,2,69,22:print"[2 down]"
     tab(39)chr$(143)"[ctrl<6>]Stampa
     [spc]File":return
1170 :
1180 window11,2,69,22:print"[3 down]"
     tab(12)chr$(15)"[ctrl<8>]Vedi
     [spc]Disk[spc]Directory[ctrl<2>]"
1190 window12,8,68,17,1:window17,9,68,
     17,1
1200 directory
1210 gosub1960
1220 window12,8,68,17,1
1230 window11,2,69,22:print"[3 down]"
     tab(12)chr$(143)"[ctrl<6>]Vedi
     [spc]Disk[spc]Directory":return
1240 :
1250 print"[home][down][cmdr<1>]
     [2 left]"chr$(15)"Calcolando...
     [spc]"
1260 print"[home][2 down][ctrl<6>]"
1270 i=1:do until a(2,i)=0
1280 a(6,i)=a(3,i)+a(4,i)+a(5,i)
1290 a(7,i)=a(6,i)/a(2,i)
1300 a(8,i)=a(8,i-1)+a(2,i)
1310 a(9,i)=a(9,i-1)+a(6,i)
1320 a(10,i)=a(9,i)/a(8,i)
1330 i=i+1:loop
1340 fori=1+pgto13+pg:forj=6to10
1350 printtab(tb(j));:printusing"#####
     ";a(j,i);
1360 nextj:print"[spc]":nexti
1370 print"[home][down][rvon][cmdr<1>]
     "chr$(143)"[16 spc]"
1380 return
1390 :
```

bowling, in questo modo si possono immettere nel programma un massimo di 52 sessioni. L'intestazione delle colonne dei dati prevede l'inserimento della data della sessione del gioco, il numero di giochi effettuati (1-3) in quella sessione e i punteggi dei giochi. Gli altri titoli di cui il programma elabora i dati, includono i pin totali, la media per sessione e il numero cumulativo delle partite effettuate. La parte inferiore dello schermo è riservata all'elenco dei vari comandi disponibili nella modalità di editing.

Quando si accede a questo libro mastro elettronico dal menù principale appare sullo schermo la prima delle quattro pagine di cui possiamo disporre. Per l'immissione dei dati si utilizzino i tasti cursore per posizionare il cursore sulla cella dove si desidera immettere o modificare un dato, poi si prema il tasto con la freccia rivolta verso l'alto. Quando appaiono la parola 'enter' e un cursore lampeggiante sulla linea di stato rossa, si digitino i dati che si vogliono introdurre nella cella e si prema il tasto Return. Si immetta la data introducendo il mese e il giorno, ad esempio per il 6 giugno si digiti 06/06. Se si preme il tasto Return senza immettere la data, i contenuti della cella rimangono invariati.

Dopo aver immesso tutti i dati e operate tutte le modifiche necessarie, si prema la freccia indicante a sinistra, in questo modo il programma elaborera' i rimanenti calcoli del foglio elettronico. Per spostarsi tra le quattro pagine a disposizione del programma si premano contemporaneamente i tasti CTRL e un numero da uno a quattro a secondo della pagina che si desidera visualizzare. In questa operazione avrete sempre visualizzato il numero della pagina in cui vi trovate e il nome del file sulla prima riga dello schermo.

Quando state analizzando una pagina del programma, sarete in grado di trovare un qualsiasi gioco tra i 200 immagazzinati e una qualsiasi serie di punteggi tra i 500 a disposizione, per ottenere ciò non dovrete far altro che premere contemporaneamente i tasti CTRL/F. Il punteggio verra evidenziato con un colore rosso.

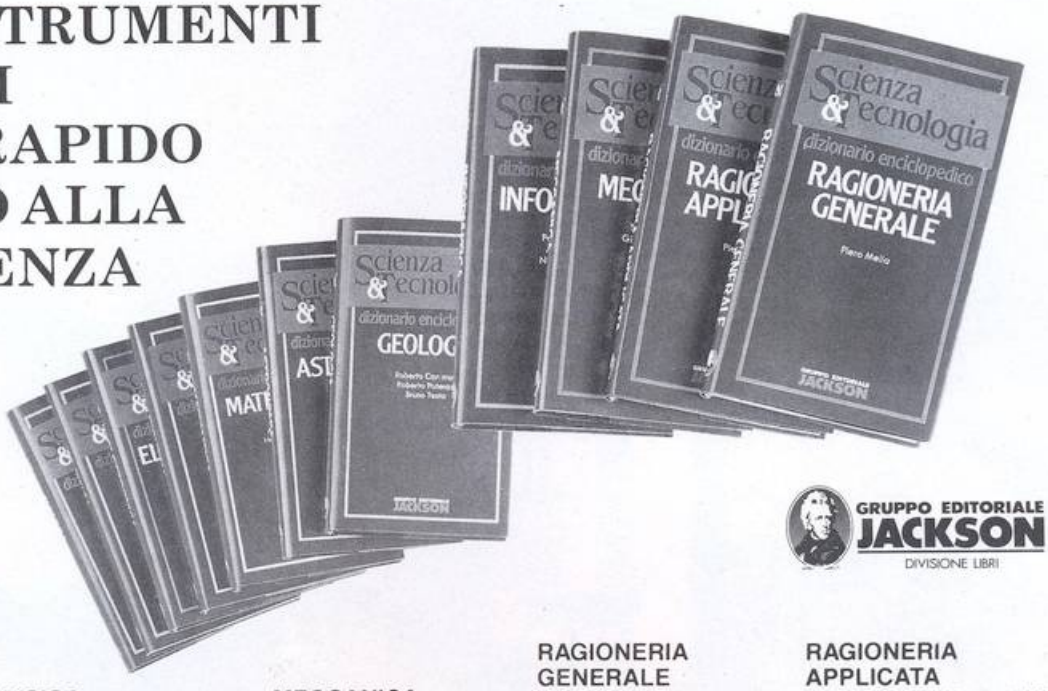
Una volta esaurito completamente un gioco e resi quindi inutili i punteggi registrati, è possibile cancellare l'intero contenuto di un file per creare un nuovo file per mezzo della pressione dei tasti CTRL/C. Ricordatevi che tutti i dati presenti in memoria andranno persi, accertatevi perciò che siano stati precedentemente registrati sul dischetto. Il programma vi inviterà con un prompt a verificare quest'ultima scelta.

All'interno dell'utilizzo di tale programma è possibile, in qualsiasi momento, rientrare nel menù principale da una delle pagine. A tale scopo sarà sufficiente premere il tasto 'M' per ritrovarsi nel menù principale pronti per una nuova scelta. Prima di concludere un'ultima informazione: le routine di calcolo e di stampa del programma si arrestano quando raggiungono lo zero nella colonna 'numero delle partite', ci si assicuri pertanto di non tralasciare nessuna riga durante l'immissione dei dati. Per poter utilizzare questo programma sono sufficienti queste poche informazioni ed è proprio questa semplicità che vi permetterà di utilizzarlo senza problemi per immagazzinare ed elaborare i dati delle vostre partite. Ed ora...buon divertimento.

```
1400 print"[home][down][cmdr<1>]
     [2 left]"chr$(15)"Cercando...
     [spc][cmdr<3>]"
1410 c1=c:r1=r
1420 fori=3to5:forj=1+pgto13+pg
1430 ifa(i,j)>199thenc=i:r=j-pg:gosub1
     880
1440 next:next
1450 forj=1+pgto13+pg
1460 ifa(6,j)>499thenc=6:r=j-pg:gosub1
     880
1470 next
1480 c=c1:r=r1
1490 print"[home][down][rvon][cmdr<1>]
     "chr$(143)"[16 spc]"
1500 return
1510 :
1520 window10,1,70,22
1530 print"[rvon][ctrl<3>][cmdr<a>]
     [22 shift<*>][ctrl<2>]B[spc]o
     [spc]w[spc]l[spc]i[spc]n[spc]g
     [ctrl<3>][23 shift<*>][cmdr<s>]"
1540 print"[rvon][shift<->][58 left]
     [shift<->]"
1550 print"[rvon][shift<->][9 left]
     [ctrl<6>][rvoff]1][spc]Load[spc]F
     ile[15 spc]2][spc]Save[spc]File
     [10 spc][ctrl<3>][rvon][shift<->]
     "
1560 print"[rvon][shift<->][9 left]
     [ctrl<6>][rvoff]3][spc]Vedi/Modif
     ica[spc]Dati[6 spc]4][spc]Stampa
     [spc]File[8 spc][ctrl<3>][rvon]
     [shift<->]"
1570 print"[rvon][shift<->][9 left]
     [ctrl<6>][rvoff]5][spc]Vedi[spc]D
     isk[spc]Directory[5 spc]6][spc]Es
     ci[15 spc][ctrl<3>][rvon]
     [shift<->]"
1580 print"[rvon][shift<->][58 left]
     [shift<->]"
1590 print"[rvon][ctrl<3>][cmdr<q>]
     [58 shift<*>][cmdr<w>]"
1600 fori=1to10:print"[rvon][shift<->]
     [58 left][shift<->]":next
1610 print"[rvon][ctrl<3>][cmdr<z>]
     [58 shift<*>][cmdr<x>]"
1620 return
1630 :
1640 r=1:c=1:f$=".tp":pg=0:p=1:sc=0
1650 fori=1to53:d$(i)="01/01":next
1660 fori=1to10:forj=1to53:a(i,j)=0:
     next:next
1670 return
1680 :
1690 v$="":k$="":ct=0:fl=1:print"
     [home][down][cmdr<1>]"tab(12);
1700 sys52684,192,10
1710 getkeyk$
1720 ifk$=chr$(13)then1840:rem return
1730 ifk$=chr$(20)andct<>0thenprint"
     [right][spc][right]";:ct=ct-1:v$=
     left$(v$,ct):goto1710
1740 ifk$="[spc]"then1810:rem spazio
1750 ifk$="."then1810
1760 ifty=1or ty>2thenif(k$>"`")and(k$
     <"[")then1810:rem alfa
1770 ifty>1thenif(k$>"/")and(k$<":")
     then1810:rem numerico
1780 ifty=4thenif(k$>"[spc]")and(k$<"0
     ")then1810:rem !"#$%&'()*+,-./ pu
     nteggiatura
1790 ifty=4thenif(k$>"9")and(k$<"`")
     then1810:rem :;<=>? punteggiatura
1800 goto1710
1810 ifct>=1then1710
1820 ct=ct+1:v$=v$+k$:print"[rvon]"k$;
1830 goto1710
1840 ifct=0thenfl=0
1850 sys52684,160,10:print"[rvoff]";:
     return
1860 :
1870 print"[ctrl<6>]";
1880 ifc=1thenchar1,c+1,r+2,"":print
     using"[rvoff]#####";d$(r+pg)
1890 ifc>1thenchar1,(c-1)*8+2,r+2,"":
     printusing"[rvoff]#####";a(c,r+pg
     )
1900 return
1910 print"[cmdr<7>]";
1920 ifc=1thenchar1,c+1,r+2,"":print
     using"[rvon]#####";d$(r+pg)
1930 ifc>1thenchar1,(c-1)*8+2,r+2,"":
     printusing"[rvon]#####";a(c,r+pg)
1940 return
1950 :
1960 print"[2 spc]"chr$(2)"[down]
     [ctrl<5>]Premi[spc]un[spc]tasto
     [spc]e[spc]continua.";chr$(130)
1970 getkeyk$
1980 return
1990 :
2000 print"[home][down][rvon][cmdr<1>]
     [2 left]Sei[spc]sicuro[spc]?"
2010 getkeyk$:ifk$<>"y"then2040
2020 print"[home][down][cmdr<1>]"chr$(
     15)"[2 left]Crea[spc]nuovo[spc]fi
     le..."
2030 gosub1640:sc=0
2040 print"[home][down][rvon][cmdr<1>]
     "chr$(143)"[20 spc]"
2050 return
2060 :
2070 sys52684,0,12:sys52684,8,20:poke2
     606,0:poke2607,8:return:rem scher
     mo 1
2080 sys52684,16,12:sys52684,24,20:
     poke2606,16:poke2607,24:return:
     rem schermo 2
10000 scratch"libro[spc]bowling":dsave"
     libro[spc]bowling"
```

Scienza & Tecnologia dizionari enciclopedici

UNDICI STRUMENTI PREZIOSI PER UN RAPIDO ACCESSO ALLA CONOSCENZA



BIOLOGIA
Cod. DS529 pp. 416 L. 14.000

Le varie branche della Biologia (botanica, zoologia, biochimica, fisiologia, immunologia e genetica) sono i temi di quest'opera che rivolge particolare attenzione anche ai più recenti risultati della biologia molecolare. Sono inoltre trattati anche i concetti delle discipline biomediche, quali patologia, istologia, farmacologia, microbiologia.

INFORMATICA
Cod. DS531 pp. 288 L. 14.000

Un acronimo di difficile comprensione, i più recenti traguardi raggiunti dall'Intelligenza Artificiale, la struttura di un personal computer. Domande ricorrenti per chi vuole conoscere una disciplina giovane che, nello spazio di pochi anni, ha cambiato il modo di produrre il nostro lavoro.

FISICA
Cod. DS498 pp. 272 L. 14.000

A molti sono forse noti gli straordinari risultati delle moderne ricerche della fisica nucleare e di quella delle particelle. Ma non altrettanto noti sono probabilmente i concetti di base della fisica moderna che hanno permesso di raggiungere questi importanti traguardi. Il dizionario di Fisica ti aiuterà a conoscerli.

MECCANICA
Cod. DS530 pp. 240 L. 14.000

Funzionamento dei meccanismi, loro progettazione e verifica, processi di produzione delle leghe, studio delle tecnologie per la lavorazione dei materiali, principi e mezzi per produrre energia, come sono realizzati gli strumenti per la rilevazione di grandezze meccaniche o fisiche. Ecco alcuni temi del dizionario di Meccanica.

RAGIONERIA GENERALE
Cod. DS527 pp. 304 L. 14.000

Sempre più rilevante è il ruolo che sta assumento la Ragioneria sia nelle scuole, sia nelle aziende, sia infine nella formazione professionale. Il primo dizionario, dedicato alla Ragioneria Generale, comprende la spiegazione dei concetti di base dei sistemi e dei metodi contabili ed è arricchito dalla terminologia relativa alle società.

RAGIONERIA APPLICATA
Cod. DS528 pp. 288 L. 14.000

Il secondo dizionario è dedicato alla Ragioneria Applicata ed include i concetti e le spiegazioni utili per lo studente, il professionista e per chi opera nelle imprese mercantili, industriali, bancarie, assicurative. Termini di contabilità degli enti pubblici, termini impiegati nelle analisi e nella formazione del bilancio completano la struttura dell'opera.

CHIMICA
Cod. DS526 pp. 304 L. 14.000

Quante volte ci siamo trovati nella necessità non solo di verificare una formula complessa, ma anche di conoscere un concetto di chimica generale, inorganica, analitica inquadrato in un contesto scientifico più ampio? Questo dizionario ti aiuterà passo a passo nella conoscenza della Chimica.

MATEMATICA
Cod. DS499 pp. 296 L. 14.000

Il dizionario di Matematica aiuterà non solo a chiarire e rinfrescare concetti magari già noti, ma anche a conoscere i progressi di questa materia che, certamente pari a quelli di altre discipline e non solo di carattere teorico, stanno assumendo un'importanza crescente nei più svariati settori applicativi.

ELETTRONICA
Cod. DS524 pp. 384 L. 14.000

Il rapido sviluppo tecnologico di questi anni è sovente causa dell'obsolescenza non solo di componenti e sistemi elettronici, ma anche di concetti, documentazione e libri che trattano questa disciplina. Una delle funzioni di questo dizionario è quella di fornire un valido sussidio per rimanere sempre aggiornati.

GEOLOGIA
Cod. DS522 pp. 288 L. 14.000

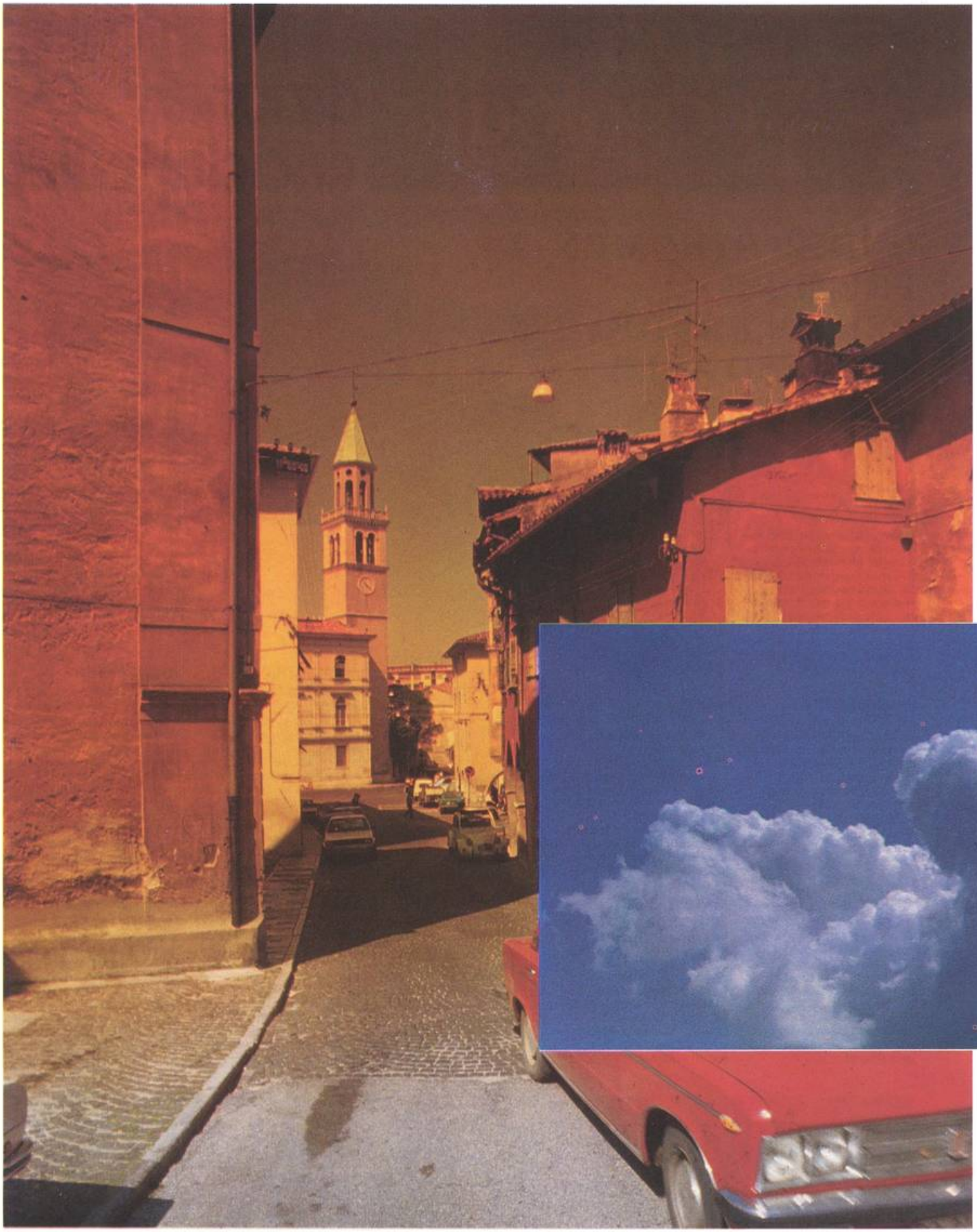
Sopresti collocare termini come magnetudo, evaporite, cratone, faglia nel contesto della natura che ci circonda? Questi e altri 1200 termini sono spiegati nel dizionario di Geologia che utilizza un completo sistema di rimandi per agevolare il lettore nella conoscenza e nell'approfondimento della materia.

ASTRONOMIA
Cod. DS525 pp. 304 L. 14.000

È esplosa una supernova nella Grande Nube di Magellano. Ma dov'è la Grande Nube e cos'è una supernova? Sono queste le domande che ci poniamo quando il cielo fa notizia sui quotidiani o nei notiziari televisivi e alle quali potremo trovare risposta in questo dizionario.

I PICCOLI GRANDI DIZIONARI JACKSON
NELLE MIGLIORI LIBRERIE

Per acquistare libri Jackson rivolgetevi alle migliori librerie e negozi di informatica oppure utilizzate l'apposito tagliando riportato in fondo alla rivista.



Inventario della casa

Difendetevi dalle perdite registrando la descrizione, la locazione e il valore degli oggetti presenti in casa

Quando il fuoco o dei ladri visitano la vostra casa, non c'è nulla al mondo che possa sollevarvi dalla sensazione di distruzione o violazione della vostra privacy; ma almeno, riuscite in parte a rassicurarvi, pensando al fatto che la vostra assicurazione vi aiuterà a coprire parzialmente la perdita finanziaria dovuta al danno subito. Se un fatto del genere vi è accaduto e di conseguenza avete richiesto il rimborso dei danni, siete senz'altro consapevoli di quanto è frustrante una simile esperienza. Anche se siete in grado di ricordarvi ogni cosa che avete perso e siete capaci di descriverla in modo dettagliato descrivendone le qualità, le condizioni o altro. L'identificazione per mezzo dei numeri di serie o dimensioni è necessaria per la Polizia per scoprire e rendere la refurtiva ai legittimi proprietari.

È possibile evitare molti di questi problemi utilizzando il programma che vi proponiamo per preparare un elenco dettagliato e sempre aggiornato delle vostre proprietà. Un catalogo di questo tipo può farvi scegliere più correttamente anche qual'è l'esatto ammontare dei beni che dovete assicurare, e vi renderà più snelle le operazioni necessarie in tali situazioni;allo stesso modo sarete in grado di riferire alla Polizia in maniera molto dettagliata sulle caratteristiche degli oggetti eventualmente a voi rubati.

Doppia versione

Il primo ed il secondo listato sono rispettivamente per il Commodore 64 ed il C-128. Entrambe le versioni sono scritte totalmente in Basic. Ora non dovete far altro che ricopiare la versione appropriata, registrarla su dischetto e poi provarla mediante il comando Run. Vi suggeriamo di dedicare un nuovo dischetto per questa applicazione, nel quale porrete il programma e i file inventario che verranno a crearsi quando registrerete i vostri dati. In questo modo qualsiasi operazione intendiate effettuare risulterà priva di difficoltà.

La versione per il Commodore 128 ha un numero maggiore di sofisticate configurazioni rispetto alla versione per il C-64. Innanzitutto, viene utilizzata con lo schermo in 80 colonne, permettendovi di analizzare con un singolo colpo d'occhio un numero superiore di informazioni mentre digitate. Si serve delle window per visualizzare le istruzioni e lo stato dell'inventario.

Inizio di un file

Quando attivate il programma, la prima schermata visualizza la situazione del vostro file, vi sono inclusi i numeri delle stanze e i numeri degli item che si possono inserire in ogni stanza, ed infine il numero degli item immessi fino a quel momento. Naturalmente, se avete appena iniziato quest'ultimo numero sarà uguale a zero. Nella parte inferiore dello schermo appare il menù principale.

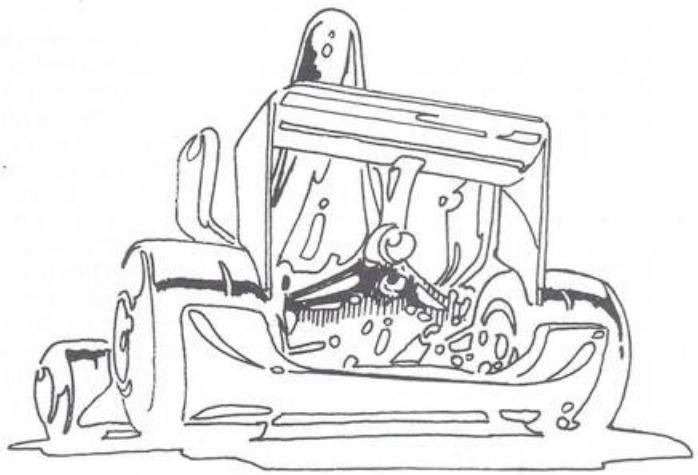
Scegliete l'opzione 1, 'Crea File'; poi quando vi verrà richiesto, inserite un nome per identificarlo che non sia più lungo di 12 caratteri. Se inserite una data, ad esempio Novembre 1988 come nome del file, vi sarà più facile in seguito rintracciare l'aggiornamento più recente sul dischetto. Qualunque sia il nome che voi intendiate dare al file, il programma vi applicherà il suffisso '.INV' (ad esempio: Novembre 1988.INV), per identificarlo come un file del programma Inventario. Se avete recensito piu' item di quanti un file può sostenerne, non dovrete far altro che espandere il vostro inventario in un secondo file.

Il programma presume che voi raggruppiate gli item concordandoli con la stanza dove tenete usualmente quegli oggetti. Dopo aver denominato un file, dovrete specificare quante stanze ha la vostra casa ed il numero massimo di

inventario 128

```
  0 rem inventario per la casa 128
  5 rem ************************
 10 nf$="nessuno":print"80[spc]colonn
    e"
 20 graphic 5:color 6,1:print"
    [2 home][clr][cmdr<7>]"
 30 for i=0 to 79:char,i,4,"C":next
 40 for i=0 to 3:char,44,i,"B":next:
    char,44,4,"[cmdr<e>]"
 50 printchr$(142):m$(0)="end":sa=7
 60 m$(1)="crea[spc]file"
 70 m$(2)="prendi[spc]file[spc]da
    [spc]disco"
 80 m$(3)="elaboro[spc]dati"
 90 m$(4)="save[spc]su[spc]disco"
100 m$(5)="rapporto[spc]stampato"
110 do:window 0,0,35,3,1:color5,15
120 printchr$(142)tab(9)"*[3 spc]inve
    ntario[3 spc]*"
130 printtab(9)"*[4 spc]c.p.[spc]1988
    [3 spc]*"
140 gosub 230:window 0,5,79,24,1
150 print"[down][spc]menu'[spc]princi
    pale[down]"
160 for i=0 to 5:printi;m$(i):next
170 print"[down]scelta?":getkey a$:a=
    val(a$)
180 fast:print"[clr]"m$(a)
190 if a>2 and nr=0 then input"no
    [spc]dati!";a$:run
200 if a>2 or nr=0 then 220
210 print"[down]sei[spc]sicuro?[spc]y
    /n":getkey a$:loop until a$="y"
220 on a+1 gosub 290,660,310,780,450,
    1580:loop
230 window 46,0,79,3,1:color5,14
240 print"nome[spc]file:[2 spc]"nf$
250 print"totale[spc]stanze[spc]",nr
260 print"item[spc]per[spc]stanza",ni
270 print"item[spc]immessi[spc]",ci;
280 return
290 print"[2 home][clr]":end
300 :
310 clr:gosub 580:if b>19 then run
320 input#1,a,b
330 print"[down]file[spc]"nf$"[spc]pe
    rmessi"a"stanze,[spc]&"b"item
    [spc]per[spc]stanza."
340 print"premi[spc]return[spc]se
    [spc]ok,[spc]o[spc]enter:"
350 i=a:j=b:gosub 700
360 :
370 for i=1 to a
380 input#1,nr$(i),ir(i)
390 for j=1 to ir(i)
400 input#1,it$(i,j),id$(i,j),ag(i,j)
    ,nv(i,j),pv(i,j)
410 next:if j>ni then ir(i)=ni
420 next
430 gosub 550:gosub 960:goto 50
440 :
450 print"press[spc]return[spc]for:
    [spc]"nf$
460 gosub 580:if b>19 then return
470 print#1,nr:print#1,ni:b$="[spc]"
480 for i=1 to nr
490 print#1,b$+nr$(i):print#1,ir(i)
500 for j=1 to ir(i)
510 print#1,b$+it$(i,j):print#1,b$+ic
    $(i,j)
520 print#1,ag(i,j):print#1,nv(i,j):
    print#1,pv(i,j)
530 next:next
540 :
550 if ds>19 then print"problema:
    [spc]"ds$:getkey a$
560 dclose:return
570 :
580 input"nome[spc]file";nf$
590 d$=left$(nf$,12)+".inv"
600 do:if a then dopen#1,(d$),w:else
    dopen#1,(d$)
610 b=ds:if b<20 then return
620 print"problema:[spc]"ds$:dclose:
    if b=63 then getkey a$:return
630 print"rimpiazzo[spc]vecchio[spc]f
    ile?[spc]y/n":getkey a$:if a$<"y"
     then return
640 scratch(d$):loop
650 :
660 clr:input"nome[spc]file";nf$
670 print"[down]gli[spc]item[spc]sono
    [spc]raggruppati[spc]per[spc]stan
    za[spc]di[spc]appartenenza."
680 gosub 700: goto50
690 :
700 input"[down]massimo[spc]#[spc]sta
    nze";i
710 input"massimo[spc]item[spc]per
    [spc]stanza";j
720 nr=i:ni=j
730 if a>i then i=a
740 if b>j then j=b
750 dim it$(i,j),id$(i,j),ag(i,j),nv(
    i,j),pv(i,j),nr$(i),ir(i)
760 return
770 :
780 do:gosub 1200
790 input"lavoro[spc]su[spc]quale
    [spc]stanza[spc](q=fine)";a$
800 a=val(a$)
810 if a=0 then do until a$=nr$(a):a=
    a+1:loop until a>nr
820 if a<1 or a>nr then exit
830 gosub 990
840 window 0,5,79,24,1:color 5,4
850 if ir(a)=0 then gosub 1020: goto8
    40
860 print"[spc]0.[spc]menu'[spc]princ
    ipale"
870 print"[spc]1.[spc]aggiungo[spc]it
    em"
880 print"[spc]2.[spc]elenco[spc]item
    "
```

```
 890 print"[spc]3.[spc]item[spc]in
     [spc]dettaglio"
 900 print"[spc]4.[spc]modifico[spc]da
     ti"
 910 print"[spc]5.[spc]un'altra[spc]st
     anza"
 920 print"scelta?":getkey a$:b=val(a$
     )
 930 if b<1 then exit
 940 scnclr:if b<5 then on b gosub 104
     0,1250,1110,1150:goto 840
 950 loop
 960 ci=0:for i=1 to nr
 970 ci=ci+ir(i):next
 980 return
 990 window 0,3,43,3,1:color 5,11
1000 print"stanza"a;nr$(a)"[spc]item:"
     ir(a);:return
1010 :
1020 if nr$(a)="" then input"nome[spc]
     stanza[spc]";nr$(a)
1030 :
1040 b=ir(a)+1:gosub 1510
1050 do:gosub 990:window 0,10,79,24,1:
     if b>ni then exit
1060 gosub 1370:if a$="q" then return
1070 ir(a)=b:b=b+1:ci=ci+1
1080 gosub 230:loop
1090 input"stanza[spc]completa";a$:
     return
1100 :
1110 do:input"dettaglio[spc]item[spc]#
     [spc](0[spc]fine)";b
1120 if b<1 or b>ir(a) then return
1130 gosub 1290:loop
1140 :
1150 print"premi[spc]return[spc]a[spc]
     ogni[spc]prompt[spc]per[spc]mante
     nere[spc]stessi[spc]dati.":window
      0,6,79,24
1160 do:input"[down]modifico[spc]quale
     [spc]item[2 spc]#[spc](0[spc]fine
     )";b
1170 if b<1 or b>ir(a) then return
1180 scnclr:gosub 1370:loop
1190 :
1200 print"stanza[spc]attivata:",,"ite
     m"
1210 for i=1 to nr
1220 printi,nr$(i)tab(32)ir(i)
1230 next:return
1240 :
1250 for i=1 to ir(a)
1260 printi;it$(a,i)
1270 next:input"[rvon]return";a$:
     return
1280 :
1290 print"[down]item[spc]#:",b
1300 print"descriz:",it$(a,b)
1310 print"ident:",id$(a,b)
1320 print"eta'[spc]anno:",ag(a,b)
1330 print"costo:",nv(a,b)
1340 print"valore:",pv(a,b)
1350 print:return
1360 :
1370 color 5,4:print"item[spc]#"b,"des
     crizione[spc](q:[spc]fine):"
1380 printtab(2)it$(a,b)
1390 char,0,1:inputa$:if a$="q" then
     return
1400 it$(a,b)=a$
1410 char,0,5:print"id:[2 spc]"id$(a,b
     )
1420 char,3,5:inputid$(a,b)
1430 char,0,9:print"eta'[spc]ann:",ag(
     a,b)
1440 char,9,9:inputag(a,b)
1450 print"costo[spc]origin.:[spc]"nv(
     a,b)
1460 char,14,10:inputnv(a,b)
1470 print"costo[spc]attuale:[spc]"pv(
     a,b)
1480 char,14,11:inputpv(a,b)
1490 return
1500 :
1510 color 5,15:print"[clr]per[spc]cia
     scun[spc]item[spc]fornire[spc]una
     [spc]descrizione[spc]del[spc]form
     ato,[spc]condizioni[spc]ecc."
1520 print"dai[spc]identificazione
     [spc]seriale[spc]#[spc]o[spc]usa
     [spc]altri[spc]simboli."
1530 print"non[spc]sono[spc]permessi
     [spc]input[spc]di[spc]virgole
     [spc]&[spc]punti."
1540 print"eta'[spc]espressa[spc]in
     [spc]anni"
1550 print"costo[spc]e'[spc]prezzo
     [spc]originale[spc]item,[spc]&
     [spc]stima[spc]l'attuale[spc]valo
     re."
1560 return
```



```
1570 :
1580 print"la[spc]stampante[spc]e'
     [spc]pronta?[down]"
1590 gosub 1200
1600 input"[down]inizio[spc]rapporto
     [spc]stanza[2 spc]1[3 right]";s
1610 print"fine[spc]rapporto[spc]a
     [spc]stanza[spc]"nr
1620 print"[up]"tab(18);:inpute:print
1630 if s<1 or s>e or e>nr then return
1640 :
1650 scnclr:print"descrizione[spc]&
     [spc]ident.[spc]brevi[spc]per
     [spc]adattarsi[spc]alla[spc]pagin
     a."
1660 print"eccovi[spc]le[spc]opzioni
     [spc]del[spc]rapporto:":print
1670 print"0.[spc]scrivi[spc]solo[spc]
     dati[spc]sommari"
1680 print"1.[spc]breve[spc]tabella
     [spc]id,[spc]eta',[spc]&[spc]cost
     o"
1690 print"2.[spc]descrizione[spc]inte
     ra[spc]&[spc]id'"
1700 print"3.[spc]rapporto[spc]complet
     o"
1710 print"4.[spc]setta[spc]indirizzo
     [spc]secondario"
1720 print"5.[spc]cancella"
1730 print"scelta?":getkey a$:m=val(a$)
1740 if m=4 then print"indirizzo[spc]s
     econdario[spc]e'"sa"per[spc]upper
     [spc]e[spc]lower[spc]case,[spc]co
     n[spc]line[spc]feed.":input"nuovo
     [spc]indirizzo[spc]secon.";sa:
     goto1650
1750 if m>4 then return
1760 :
1770 open4,4,sa:cmd4:print
1780 printchr$(14)"[2 spc]inventario
     [2 spc]"
1790 print"file:[spc]"nf$
1800 t1=0:t2=0:t3=0:r$="###":s$=r$+r$:
     t$=s$+s$+r$:u$=t$+t$+t$+"#"
1810 for a=s to e:i=0:j=0
1820 print:printchr$(14)"stanza"a;nr$(
     a):printchr$(15)
1830 if ir(a) then begin
1840 if mand1 then print"[2 spc]#[spc]
     descrizione"spc(35)"id"spc(13)"et
     a'[spc]costo[spc]val."
1850 for b=1 to ir(a)
1860 i=i+nv(a,b):j=j+pv(a,b):on m
     gosub 2050,2070,2030
1870 next:bend
1880 :
1890 print"totale[spc]per[spc]stanza";
     :print using r$;a;
1900 if mand1 then printspc(51);:print
      using s$;i,j:else print:b=ir(a):
     gosub 1980
1910 t1=t1+b:t2=t2+i:t3=t3+j
1920 next:print
1930 :
1940 if e-s then begin
1950 printchr$(14)"totale[spc]stanze
     [spc]"s"to"e:printchr$(15)
1960 b=t1:i=t2:j=t3:gosub 1980:bend
1970 print#4:close4:return
1980 print"items:"spc(9)b
1990 print"costo[spc]originale:[spc]"i
2000 print"costo[spc]attuale:[spc]"j
2010 return
2020 :
2030 if len(it$(a,b))>46 then printit$
     (a,b)
2040 if len(id$(a,b))>15 then printid$
     (a,b)
2050 print using r$;b;:print"[spc]";:
     print using u$;it$(a,b);:print
     using t$;id$(a,b);:print using r$
     ;ag(a,b);:print using s$;nv(a,b),
     pv(a,b)
2060 return
2070 printit$(a,b):printid$(a,b):print
     :return
```

item per stanza. Il numero di item che il vostro computer può manipolare dipende dal fatto se siete più o meno dei grafomani. In linea di massima il Commodore 64 può manipolare circa 300 item (sette stanze, con 40 item ciascuna) e il C-128 ha una capacità che raggiunge tranquillamente il doppio di quella del C-64.

In alcuni casi, potreste voler definire stanze dei luoghi che non sono dei veri e propri spazi fisici. Ad esempio, potreste definire la vostra superba collezione di vecchi dischi come una stanza, dal momento che non potrebbe esserci spazio a sufficienza per elencare ogni singolo disco all'interno della voce 'salotto'.

Dopo che avrete specificato il numero delle stanze e gli item, riapparirà lo schermo 'stato del file/menù' principale', con la linea di stato che vi indicherà l'esistenza del nuovo file.

Se avete già creato e registrato precedentemente un file su disco, sarà possibile caricarlo in memoria per aggiornarlo selezionando dal menù principale l'opzione 2, 'Prendi File da Disco'. Il programma vi darà la possibilità di effettuare delle modifiche per quanto riguarda i limiti degli item. Per mantenere la stessa ampiezza del file, basterà premere il tasto Return all'apparire dei vari prompt. Se decidete di ridurre la grandezza del file, fate attenzione perché si possono perdere alcuni item di dati durante questa operazione.

Mentre utilizzate il programma, ricordatevi che le opzioni del menù principale riguardanti l'atto di creare un nuovo file, caricare un file da disco (opzione 2), e fine del programma (opzione 0) cancellano i dati presenti in memoria. Per questa ragione, se selezionate una di queste opzioni, vi verrà chiesto di confermare le vostre intenzioni prima che il programma esegua la routine scelta. Presumendo che abbiate una copia dei dati su dischetto si prema 'Y' e il tasto Return per informare il programma che può procedere nelle operazioni.

Stesura di un inventario

Dopo aver terminato di creare un file o averne caricato uno da disco, si scelga l'opzione 3 del menù principale 'Elaboro Dati' e il programma visualizzerà l'elenco delle stanze e il numero di item descritti per ciascuna di esse. Per

selezionare una stanza da inventariare, si può immettere o il suo numero o il suo nome. Dopo aver selezionato una stanza si accede ad un secondo menù. Per aggiungere gli item si scelga l'opzione 1.

Se attivate una nuova stanza, dovrete fornirla di un nome, poi iniziate a immettere gli item, in questo modo accederete al secondo menù. Ora potrete entrare nei dettagli di ogni singolo item. Analizzate la stanza (quella reale) e immettete ogni singolo oggetto (gli item più piccoli possono essere raggruppati ed immessi come uno solo). Per ciascun item fornite una completa descrizione incluso il tipo di materiale con cui è costituito, il modello, la misura, il colore, le condizioni in cui si trova, eccetera. Potete immettere due linee di schermo (80 caratteri per il Commodore 64 e 160 caratteri per il C-128).

L'unico limite alle vostre descrizioni, dal momento che il programma è scritto in Basic, è che non potete inserire virgole o punti; comunque i punti e virgola o il trattino sono dei separatori altrettanto validi. È possibile raggirare questo impedimento iniziando l'input con delle virgolette, ma questo rende la descrizione più difficile da editare.

Poi immettete un'unica caratteristica identificatrice, ad esempio un numero seriale.

I restanti dati da immettere includono l'età approssimata dell'oggetto espressa in anni, il prezzo da voi pagato al momento del suo acquisto e il suo valore attuale. La vostra compagnia di assicurazione avrà bisogno di tutti questi dati per rispondere alle vostre richieste.

Per quanto riguarda il determinare il valore attuale si può procedere in due modi. Usualmente si introduce il valore del bene deprezzato. Ad esempio, un oggetto utilizzato per alcuni anni che si presenti ancora in buono stato può essere valutato la metà del suo valore originale. Se invece i valori, nella vostra polizza assicurativa vengono valutati come sostituzione dell'oggetto mancante, il valore attuale dell'oggetto verrà definito come il costo di un nuovo item che sia paragonabile a quello perso. Nell'utilizzo del programma potete utilizzare uno qualsiasi dei due metodi.

Quando avete terminato con una stanza, premete il tasto 'Q' al posto della descrizione dell'item per ritornare al menu' principale. È sempre possibile ritornare sui propri passi in un momento successivo ed aggiungere altri item finché la stanza non sia piena.

Rivedere i propri dati

Se desiderate effettuare delle aggiunte o delle modifiche ai dati già immessi, non dovete far altro che digitare il numero della stanza o il suo nome, poi scegliere l'opzione che desiderate. L'opzione uno (1) viene utilizzata per aggiungere item, mentre l'opzione due (2) visualizza la descrizione di tutti gli item attualmente residenti nella stanza. Gli item scorreranno abbastanza velocemente, perciò se intendete rallentare o bloccare il listing utilizzate gli appropriati tasti (CTRL sul C-64 e il tasto Commodore sul C-128). Si prema il tasto Return alla fine per rivedere l'elenco.

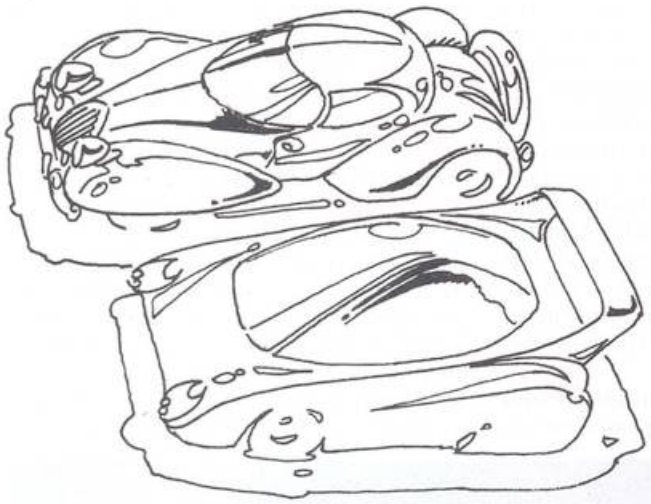
Per vedere maggiori dettagli di uno specifico item, si scelga l'opzione tre (3) del menù e si immetta il numero dell'item. Per ritornare al menù si immetta il numero zero (0) alla richiesta del numero dell'item.

Se volete modificare alcune informazioni dell'item, si scelga l'opzione quattro (4) del menù, 'Modifica Dati', e si immetta il numero dell'item. Dopo che il programma avrà elencato i dati presenti per quell'item, potete digitare le modifiche opportune; si prema il tasto Return se non si vuole modificare i dati. Quando l'operazione è terminata, si immetta il numero zero (0) come numero dell'item per ritornare al menù.

Una volta completati tutti i cambiamenti desiderati nella stanza, si scelga l'opzione cinque (5) per spostarsi in un'altra stanza e l'opzione zero (0) per ritornare al menù principale.

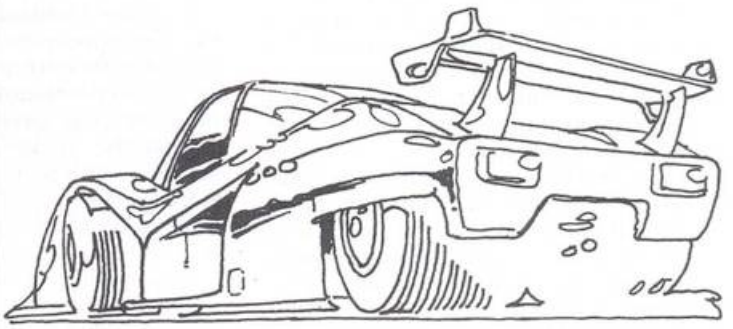
Registrare su disco

Mentre formate il vostro inventario, assicuratevi di registrare frequentemente il file su disco, in modo da non perdere ore del vostro lavoro se il computer si blocca o se c'è una momentanea assenza dell'erogazione dell'energia elettrica. Per registrare un file d'inventario si scelga dal menù principale l'opzione quattro (4). Quando vi verrà richiesto il nome del file, premete il tasto Return se intendete registrarlo con lo stesso nome. Se dediderate modificare il nome del file, digitate semplicemente il nuovo nome, e il nome verrà modificato. Se digitate un nome precedentemente utilizzato sul dischetto, il programma vi informerà che il file è già esistente sul dischetto e vi chiederà se intendete sostituirlo. Se sì, premete il tasto 'Y' e successivamente quello di Return per cancellare il vecchio file e registrare il nuovo al suo posto. Altrimenti, immettete 'N' e riprovate, utilizzando un nome diverso.



```
0 rem inventario 64 per la casa
5 rem ************************
10 nf$="nessuno"
20 print"[shift clr][crsr giu'] *
    inventario casa"
30 print"[2 crsr giu'"
40 print"nome file:",nf$
50 print"totale stanze ",nr
60 print"item per stanza",ni
70 print"item immessi ",ci
80 print"[2 crsr giu']menu' principale
   [crsr giu']"
90 sa=7:k=198:m$(0)="end"
100 m$(1)="crea file"
110 m$(2)="prendi file da disco"
120 m$(3)="elaboro dati"
130 m$(4)="save su disco"
140 m$(5)="rapporto stampato"
150 for i=0 to 5
160 printi;m$(i):next
170 print"[crsr giu']scelta?":wait k,7:
    get a$:a=val(a$)
180 print:print"[shift clr]"m$(a)
190 if a>2 and nr=0 then input
    "no dati!";a$:run
200 if a>2 or nr=0 then 220
210 print"[crsr giu']sei sicuro? y/n":
    wait k,7:get a$:if a$<"y" then 20
220 on a+1 gosub 230,690,250,820,410,
    1690:goto 20
230 end
240 :
250 clr:d$=",s,r"
260 gosub 560:if a>19 then run
270 input#1,a,b
280 print"[crsr giu']file "nf$
    " permessi"a
290 print"stanze, &"b"item per stanza."
300 print"premi return se ok, o enter:"
310 i=a:j=b:gosub 740
320 :
330 for i=1 to a
340 input#1,nr$(i),ir(i)
350 for j=1 to ir(i)
360 input#1,it$(i,j),id$(i,j),ag(i,j),
    nv(i,j),pv(i,j)
370 next:if j>ni then ir(i)=ni
380 next
390 gosub 520:gosub 1650:goto 20
400 :
410 print"premi return per: "nf$
420 d$=",s,w"
430 gosub 560:if a>19 then return
440 print#1,nr:print#1,ni:b$=" "
450 for i=1 to nr
460 print#1,b$+nr$(i):print#1,ir(i)
470 for j=1 to ir(i)
480 print#1,b$+it$(i,j):print#1,b$+id$
    (i,j)
490 print#1,ag(i,j):print#1,nv(i,j):
    print#1,pv(i,j)
500 next:next
510 :
520 close1:input#2,a,b$:close2
530 if a>19 then print"problema: "b$:
    input"[ctrl 9]return";a$
540 return
550 :
560 input"nome file";nf$
570 df$=left$(nf$,12)+".inv"
580 d$="0:"+df$+d$
590 open2,8,15
600 open1,8,8,d$
610 input#2,a,a$
620 if a<20 then return
630 printa$:close1:close2
640 if a=63 then input"[ctrl 9]return";
    a$:return
650 print"rimpiazzo vecchio file? y/n":
    wait k,7:get a$
660 if a$<"y" then return
670 open2,8,15,"s0:"+df$:goto 600
680 :
690 clr:input"[crsr giu']nome file";nf$
700 print"[crsr giu']gli item sono
     raggruppati"
710 print"per stanza di appartenenza."
720 gosub 740:goto 20
730 :
740 input"[crsr giu']massimo # di
     stanze";i
750 input"massimo item per stanza";j
760 nr=i:ni=j
770 if a>i then i=a
780 if b>j then j=b
790 dim it$(i,j),id$(i,j),ag(i,j),nv(i,
    j),pv(i,j),nr$(i),ir(i)
800 return
810 :
820 gosub 1200
830 input"lavoro su quale stanza (q=
    fine)";a$
840 a=val(a$):if a then 880
850 for i=1 to nr
860 if a$=nr$(i) then a=i
870 next
880 if a<1 or a>nr then 1650
```

```
890 if ir(a)=0 then gosub 1020
900 print:print"room"a,nr$(a),"items:"
    ir(a)
910 print" 0. menu' principale"
920 print" 1. aggiungi item"
930 print" 2. elenca item"
940 print" 3. item in dettaglio"
950 print" 4. modifica dati"
960 print" 5. un'altra stanza"
970 print"scelta?":wait k,7:get a$:b=
    val(a$)
980 if b<1 then 1650
990 if b>4 then 820
1000 print:on b gosub 1040,1250,1100,
     1140:goto 900
1010 :
1020 if nr$(a)="" then input"nome
      stanza";nr$(a)
1030 :
1040 b=ir(a)+1:gosub 1560
1050 if b>ni then input"stanza completa:
      [ctrl 9]return";a$:return
1060 gosub 1400
1070 if a$="q" then return
1080 ir(a)=b:b=b+1:goto 1050
1090 :
1100 input"dettaglio item # (0 fine)";b
1110 if b<1 or b>ir(a) then return
1120 gosub 1310:goto 1100
1130 :
1140 print"[crsr giu']premi return ad
      ogni prompt "
1150 print"per mantenere stessi dati.":
     print
1160 input"modifica quale item #
      (0 fine)";b
1170 if b<1 or b>ir(a) then return
1180 gosub 1400:goto 1140
1190 :
1200 print:print"stanze",,,"item"
1210 for i=1 to nr
1220 printi,nr$(i)tab(32)ir(i)
1230 next:return
1240 :
1250 print"[shift clr]"
1260 for i=1 to ir(a)
1270 printi;it$(a,i)
1280 next:if ir(a)>16 then input"[ctrl
      9]return";a$
1290 return
1300 :
1310 print"[crsr giu']stanza:",nr$(a)
1320 print"item #:",b
1330 print"descriz:",it$(a,b)
1340 print"ident:",id$(a,b)
1350 print"eta', anno:",ag(a,b)
1360 print"costo:",nv(a,b)
1370 print"valore:",pv(a,b)
1380 print:return
1390 :
1400 print"[shift clr]stanza:",nr$(a)
1410 print"item #"b"descrizione (q:
      per fine):"
1420 printtab(2)it$(a,b)
1430 print"[home][crsr giu']":inputa$
1440 if a$="q" then return
1450 it$(a,b)=a$
1460 print"[home][7 crsr giu']id:  "id$
     (a,b)
1470 print"[home][7 crsr giu']
     [3 crsr destra]";:inputid$(a,b)
1480 print"eta', anno: "ag(a,b)
1490 print"[crsr su]"tab(11);:inputag(
     a,b)
1500 print"costo originale: "nv(a,b)
1510 print"[crsr su]"tab(16);:inputnv(
     a,b)
1520 print"valore attuale: "pv(a,b)
1530 print"crsr su]"tab(15);:inputpv(
     a,b)
1540 return
1550 :
1560 print"[shift clr]per ogni item,
      dai una descrizione"
1570 print"incluso formato, condizione
      ecc. "
1580 print"identifica con seriale # o
      altri simb."
1590 print"non inserire virgole &
      punti."
1600 print"[crsr giu']eta' espressa ir
      anni."
1610 print"costo e' prezzo originale
      dell'item,"
1620 print"& stima il valore attuale."
1630 print"[crsr giu']premi spazio ":
     wait k,7:get a$
1640 :
1650 ci=0:for i=1 to nr
1660 ci=ci+ir(i):next
1670 return
1680 :
1690 print"[crsr giu']la stampante e'
      pronta?"
1700 gosub 1200
1710 input"[crsr giu']inizio rapporto
      stanza 1[3 crsr sn]";s
```



```
1720 print"fiñe rapporto stanza "nr
1730 print"[crsr su]"tab(18);:inpute:
     print
1740 if s<1 or s>e or e>nr then return
1750 :
1760 print"[shift clr]descrizione & id
      brevi per adattarsi"
1770 print"alle vostre opzioni:":print
1780 print"0. sommario dati"
1790 print"1. breve tabella id, eta' &
      costi"
1800 print"2. descrizione piena & id'"
1810 print"3. rapporto intero"
1820 print"4. setta indirizzo
      secondario"
1830 print"5. cancella"
1840 print"scelta?":wait k,7:get a$:m=
     val(a$)
1850 if m<4 then 1900
1860 if m>4 then return
1870 print"secondary address is"sa"for
      upper/lower"
1880 print"case, & line feeds:":input
     "nuovo valore";sa:goto 1760
1890 :
1900 open4,4,sa:cmd 4:print:print
1910 printchr$(14)"  inventario  "
1920 print"file: "nf$
1930 t1=0:t2=0:t3=0:b$="[15 spazi]"
1940 for a=s to e:i=0:j=0
1950 print:printchr$(14)"stanza"a;nr$
     (a):printchr$(15)
1960 if ir(a)=0 then 2020
1970 if mand1 then print"  # descrizion
     e"spc(35)"id"spc(13)"eta'  costo
     [2 spazi]val."
1980 for b=1 to ir(a)
1990 i=i+nv(a,b):j=j+pv(a,b):on m gosub
     2180,2220,2160
2000 next
2010 :
2020 print"totale, stanza"a
2030 b=ir(a):gosub 2110
2040 t1=t1+b:t2=t2+i:t3=t3+j
2050 next:print
2060 :
2070 if s=e then 2100
2080 printchr$(14)"totale per stanze"s"
     to"e:printchr$(15)
2090 b=t1:i=t2:j=t3:gosub 2110
2100 print#4:close4:return
2110 printb$"item:"spc(8)b
2120 printb$"costo originale:"i
2130 printb$"valore attuale:"j
2140 return
2150 :
2160 if len(it$(a,b))>46 then printit$(
     a,b)
2170 if len(id$(a,b))>15 then printid$(
     a,b)
2180 printright$(b$+str$(b)+" ",4)left$
     (it$(a,b)+b$+b$+b$,46);
2190 printleft$(id$(a,b)+b$,15)right$(
     b$+str$(ag(a,b)),3);
2200 printright$(b$+str$(nv(a,b)),6)
     right$(b$+str$(pv(a,b)),6)
2210 return
2220 printit$(a,b):printid$(a,b):print:
     return
```

Stampa dei rapporti

Dopo che avete registrato l'inventario su dischetto è possibile ottenere dei rapporti di vario tipo su carta utilizzando l'opzione cinque (5). Quando selezionate quest'opzione, il programma vi chiede se volete una stampa su carta iniziando dala prima stanza. Se sì, basterà premere il tasto Return; altrimenti, si prema il numero della stanza da cui si vuole iniziare. Poi il programma vi chiede se la stampa dovrà essere effettuata includendo fino all'ultima stanza dell'elenco. Se sì, si prema nuovamente il tasto Return; altrimenti si prema il numero dell'ultima stanza che intendete includere nell'elenco. Dopo aver indicato le proprie scelte si accederà al menù stampa.

Ci sono varie combinazioni di stampanti e interfacce del mondo Commodore, e il programma è stato redatto in modo da poter utilizzarne il più possibile. Se incontrate delle difficoltà, ad esempio spaziature doppie o presenza di caratteri strani, provate nuovamente modificando l'indirizzo secondario, utilizzando l'opzione quattro (4) del menù stampa. Il manuale della vostra stampante o interfaccia vi informerà sui valori da utilizzare per caratteri in lower o uppercase in presenza di line feed.

Il programma può generare vari tipi di rapporti. Un rapporto di Sommario Dati fornisce il numero di item presenti in ciascuna stanza, il loro costo totale e il loro valore attuale, e i dati totali per quella specifica serie di stanze. Questo tipo di informazioni sono utili per decidere eventuali modifiche nel rapporto con la vostra assicurazione.

La tabella contenente Brevi ID, Età, e Costi fornisce una breve descrizione e il valore finanziario di tutti gli item. Nel costruire questa tabella vi troverete probabilmente in imbarazzo nel formulare le descrizioni visti i limiti che vi impone. Non preoccupatevi dal momento che anche le brevi descrizioni saranno sufficienti per completare questo rapprto, che vi aiuterà a stimare i vostri reali bisogni assicurativi man mano che le vostre proprietà cresceranno.

Quando si seleziona il rapporto di Descrizione Completa e ID, i dati finanziari su item individuali vengono tralasciati e solo le descrizioni dettagliate, le caratteristiche identificative e i sommari delle stanze vengono stampati. Questo rapporto potrà servire alla Polizia per identificare e recuperare i vostri oggetti.

Un rapporto completo include una Descrizione Completa e le ID di ciascun item, la sua Età e il suo Valore. Questa stampa utilizzerà due linee complete per le descrizioni e le ID. Questa è l'opzione che vi permette di avere su foglio una descrizione completa delle vostre proprietà. Quando la stampante ha terminato il suo compito il programma ritornerà al menù principale.

Considerazioni finali

Quando avete terminato il vostro inventario, fate una copia che includa sia il programma che i file dati e fate una copia scritta dell'elenco completo degli item in vostro possesso e immagazzinate il tutto in un posto sicuro. Probabilmente sarete tentati di rimandare l'esecuzione di un compito che si preannuncia lungo e noioso. Comunque, siate certi che al momento necessario è molto più difficile fornire un elenco preciso senza avere sotto mano un elenco scritto completo in ogni suo dettaglio. Perciò digitate il programma e compilate il vostro inventario degli oggetti presenti in casa.

LABORATORIO SCUOLA

SOFTWARE DIDATTICO PER LA SCUOLA MEDIA SUPERIORE

I. Perlasca / S. Sgrignoli
ONDE E OSCILLAZIONI
Teoria, esercizi e simulazioni
MS DOS Cod. SD563I
pp. 216 • L. 50.000

M. Fusilli / M. Polvani
L'ELETTRONICA DEL COMPUTER VOL. I
Teoria, esercizi e simulazioni
MS DOS Cod. SD298I
pp. 236 • L. 50.000

M. Berretta
IDRAULICA
Teoria, esercizi e simulazioni
MS DOS Cod. SD296I
C64 Cod. 296DCM
APPLE Cod. 296DAP
pp. 144 • L. 50.000

G. Dariano / A. Marzuoli
MATEMATICA FINANZIARIA E ATTUARIALE
Teoria, esercizi e simulazioni
MS DOS Cod. SD562I
pp. 120 • L. 50.000

G. Butti
TECNICA COMMERCIALE
Teoria, esercizi e simulazioni
MS DOS Cod. SD580I
pp. 170 • L. 50.000

D. Blight / T. Shafto
MICROECONOMIA
Teoria, esercizi e simulazioni
MS DOS Cod. SD560I
C64 Cod. 560DCM
pp. 189 • L. 50.000

F. Cecconi
ESERCIZI ED ESAMI CON IL PERSONAL COMPUTER
MS DOS Cod. SD561I
pp. 180 • L. 50.000

M. Fusilli / M. Polvani
L'ELETTRONICA DEL COMPUTER VOL. II
Teoria, esercizi e simulazioni
MS DOS Cod. SD299I
pp. 168 • L. 50.000

F.P. Tramontano
ALGEBRA
Teoria, esercizi e simulazioni
MS DOS Cod. SD292I
C64 Cod. 292DCM
M20 Cod. 292DOL
pp. 120 • L. 50.000

B. Rinaldi / F.P. Tramontano
ANALISI
Teoria, esercizi e simulazioni
MS DOS Cod. SD293I
C64 Cod. 293DCM
M20 Cod. 293DOL
pp. 224 • L. 50.000

A. Martinelli / M. Persico
CHIMICA
Teoria, esercizi e simulazioni
MS DOS Cod. SD295I
C64 Cod. 295DCM
pp. 192 • L. 50.000

F. Wattemberg / M. Wattemberg
STRUMENTI DI ANALISI E CALCOLO NUMERICO
Teoria, esercizi e simulazioni
MS DOS Cod. SD549I
pp. 128 • L. 50.000

F. De Michele / M. Rosa Clot
PROBABILITÀ STATISTICA E TERMODINAMICA
Teoria, esercizi e simulazioni
MS DOS Cod. SD291I
C64 Cod. 291DCM
APPLE Cod. 291DAP
pp. 160 • L. 50.000

A. Conti / M. Rosa - Clot
MECCANICA
Teoria, esercizi e simulazioni
MS DOS Cod. SD290I
C64 Cod. 290DCM
APPLE Cod. 290DAP
pp. 300 • L. 50.000

S. Faetti / L. Fronzoni
COMPORTAMENTO DEI SISTEMI COMPLESSI: ORDINE E CAOS
Teoria, esercizi e simulazioni
MS DOS Cod. SD297I
C64 Cod. 297DCM
pp. 160 • L. 50.000

B. Rinaldi / F.P. Tramontano / D. Tognon
GEOMETRIA ANALITICA
Teoria, esercizi e simulazioni
MS DOS Cod. SD294I
C64 Cod. 294DCM
M20 Cod. 294DOL
pp. 150 • L. 50.000

M. Rosa-Clot
LA FISICA CON IL COMPUTER: LA DINAMICA
Teoria, esercizi e simulazioni
MS DOS Cod. SD225I
C64 Cod. 225DCM
APPLE Cod. 550A
pp. 270 • L. 50.000

E. Boni / S. Schacherl
FISICA ATOMICA
Teoria, esercizi e simulazioni
MS DOS Cod. SD300I
C64 Cod. 300DCM
APPLE Cod. 300DAP
pp. 168 • L. 50.000

L. Danusso / S. Schacherl
CAMPI ELETTRICI E MAGNETICI
Teoria, esercizi e simulazioni
MS DOS Cod. SD301I
C64 Cod. 301DCM
APPLE Cod. 301DAP
pp. 96 • L. 50.000

E. Rapella
ELEMENTI DI BASE DI PROBABILITÀ E STATISTICA
Teoria, esercizi e simulazioni
MS DOS Cod. SD579I
C64 Cod. 579DCM
pp. 144 • L. 50.000

GRUPPO EDITORIALE JACKSON

Nuovi strumenti per una scuola che cambia

COMPILARE E SPEDIRE IN BUSTA CHIUSA A:
GRUPPO EDITORIALE JACKSON - VIA ROSELLINI 12 - 20124 MILANO

INDICARE CHIARAMENTE CODICE, QUANTITÀ E PREZZO DEI VOLUMI RICHIESTI

Codice	Quantità	Prezzo		Codice	Quantità	Prezzo

Ordine minimo L. 30.000 + L. 4.000 per contributo spese spedizione

☐ Pagherò al postino l'importo di L. _____
Al ricevimento dell'opera.

☐ Allego assegno n. _____ di L. _____
della Banca _____

☐ Richiedo l'emissione della fattura (formula riservata alle aziende) e comunico il numero di Partita IVA _____

DATA _____ FIRMA _____

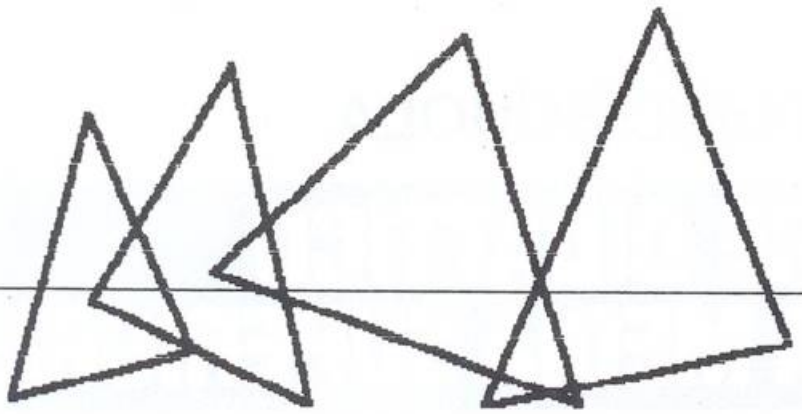
COGNOME _____

NOME _____

VIA E NUMERO _____

CAP _____ CITTÀ _____

J0116



La notte di Natale

The Night Before Christmas, è un dischetto musicale e animato per il C-64, che è stato realizzato da Simon & Schuster Software. Il programma narra la storia della natività accompagnata da un'animazione e da un sottofondo musicale.

Nel dischetto sono incluse molte altre configurazioni. Con queste potrete inviare ai vostri amici dei dischetti 'natalizi' personalizzati; digitare e stampare delle lettere per Babbo Natale (le lettere verranno indirizzate automaticamente al corretto recapito e cioè il Polo Nord); stampare dei biglietti natalizi e giocare al gioco di Natale (un puzzle computerizzato). Il dischetto è disponibile al prezzo di $8.95 e richiede una stampante MPS-801 o MPS-1000 o delle stampanti compatibili.

Per maggiori informazioni rivolgersi a Simon & Schuster Software, One Gulf + Western Plaza, New York, NY 10023, USA.

Create a Calendar

La Epyx, nota a tutti gli appassionati di video giochi come una tra le case più attive nella realizzazione di game dalla grafica altamente superba, ha messo a frutto questa sua capacità realizzando un programma per costruire dei calendari altamente personalizzabili. Il programma permette di produrre diversi calendari per gli scopi più disparati: per la casa, per la scuola o per uso ufficio.

Il programma include diversi tipi di testo, grafica e bordature; nella sezione grafica sono previsti dei disegni concernenti gli appuntamenti dentistici, i compleanni, appuntamenti dalla parrucchiera, il capodanno, la festa della mamma e moltissime altre occasioni. Possono essere utilizzati per il disegno, i dischetti grafici del Print Shop e i dischetti della collezione di Graphic Scrapbook della Epyx.

È possibile produrre dei calendari giornalieri, settimanali, mensili o annuali e i formati vanno da quello tipo lettera al tipo a sei pagine annuale.

Il programma è disponibile per il C-64 al prezzo di $29.95 presso la Epyx, P.O. Box 8020, Redwood City, CA 94063, USA.

Cad 64/128

La Weaselgraphics ha realizzato vari dischetti CAD per il C-64 e il Commodore 128, il prezzo è stato stabilito a $20 l'uno.

Con i programmi, ingegneri e hobbisti con sufficienti conoscenze tecniche saranno in grado di disegnare oscillatori professionali, amplificatori, circuiti elettrici, filtri, ecc.

I programmi possono essere utilizzati per progettare nuovi dispositivi, modificarne dei vecchi e fornire un'assistenza adeguata nelle riparazioni.

I programmi sono disponibili presso la Weaselgraphics, 606 Thomasville Road, Pocahontas, AR 72455, USA.

Basic 8.0

Un nuovo linguaggio grafico per stabilire le capacità del C-128.

Il Basic 8.0 è l'illustrazione grafica del fatto che il Commodore 128 può essere realmente programmato per eseguire delle schermate colorate in alta risoluzione sullo schermo in 80 colonne.

Il programma aggiunge 53 comandi al linguaggio di programmazione del vostro c-128, il Basic 8,0 fornisce il vostro computer di quello che viene definifito "Il sistema grafico più produttivo e potente sviluppato finora per macchine ad 8 bit".

Ed è anche una buona notizia per coloro che si sentivano defraudati dal fatto che la Commodore aveva utilizzato fino ad oggi per i display di grafica in hi-res il chip VIC, che opera esclusivamente in modalità 40 colonne.

Senza rallentare l'esecuzione delle altre operazioni in Basic, il Basic 8.0 vi fornisce il codice che la Commodore ha tralasciato nell'installare il linguaggio macchina ed effettua il patch della routine di syntax error del C-128. Il programma cerca il simbolo (′), che precede ognuno dei nuovi 53 comandi. Se l'apparente syntax error inizia con il simbolo (′), allora passa all'interprete a parte del Basic 8.0 per la processazione.

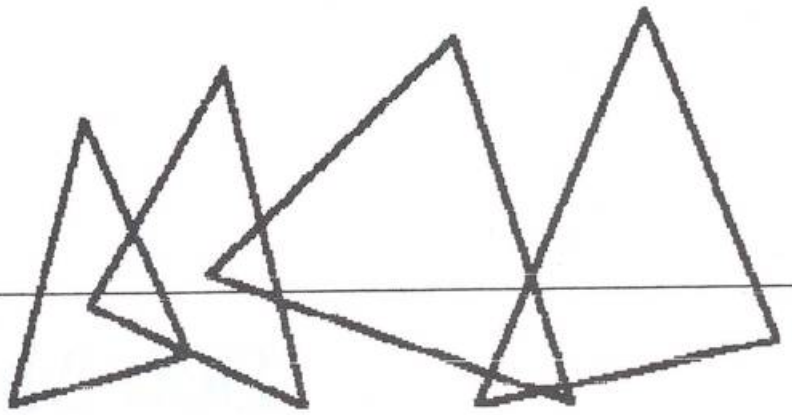
Il Basic 8.0 è completamente compatibile con il Basic 7.0 e vi permette di di programmare grafica in 80 colonne ad una risoluzione di 640×200 pixel, che è il doppio di quella di uno schermo in 40 colonne del C-128.

Il pacchetto del Basic 8.0 riserva delle ulteriori sorprese. Il singolo dischetto contiene tre moduli indipendenti: Il sofisticato linguaggio di programmazione, un modulo di run-time e un programma di disegno. In aggiunta, gli autori forniscono dei programmi di utility per convertire delle immagini create con dei loro precedenti programmi.

Se analizzato dal punto di vista del linguaggio di programmazione, i comandi del Basic 8.0 vi permettono l'accesso a nuovi modi di creare, immagazzinare, e manipolare delle immagini a colori in alta risoluzione sul vostro C-128. Il programma definisce sistemi grafici che utilizzano strutture, che vengono applicate per ogni disegno, configurazione, font o istruzioni che creerete e immagazzinerete per utilizzi futuri, e questo anche all'interno di altri programmi.

È possibile definire dei modelli colorati e bidimensionali. Il Basic 8.0 permette la realizzazione di immagini pluridisegnate mediante l'incremento dei parametri di altezza della struttura. È possibile utilizzare questo linguaggio per entrare nel mondo tridimensionale. Il programma vi fornisce un ambiente all'interno del quale è possibile controllare la prospettiva, la rotazione, l'angolo di visualizzazione la vista iniziale e finale.

Il manuale è molto chiaro e illustra ogni nuovo comando offrendo allo stesso tempo molti programmi d'esempio introducendovi in questo modo alla conoscenza di questo linguaggio più evoluto. I comandi includono: 'STASH, che vi permette di immagazzinare vari schermi grafici nei banchi della RAM 0 e 1; 'SCROLL, che sposta l'immagine in una delle otto direzioni possibili; 'HCOPY, che invia alla stampante le schermate grafiche prodotte; 'FONT, che vi permette di scegliere il set di ca-



ratteri più opportuno disponibile nel modo testo.

Altri comandi inclusi sono: 'WINDOWOPEN per aprire una finestra all'interno della quale il Basic 8.0 può continuare ad operare; 'FLASH, che effettua il reverse dell'immagine; 'ZOOM, per analizzare in dettaglio la grafica sullo schermo. Il comando 'SCALE vi permette di controllare la non simmetria di alcune forme.

Per essere sicuri che un programma scritto in Basic 8.0 operi su un qualsiasi C-128, gli autori, hanno introdotto il sistema run-time che può essere ceduto o venduto dai possessori del Basic 8.0. Questo significa che sarete liberi di vendere i programmi da voi creati con il Basic 8.0 senza dover pagare i diritti d'autore.

Un'altra parte di questo pacchetto consiste nel Basic Paint, un programma di disegno con molteplici funzioni. Tale programma vi offre un ricchissimo tool per creare schermate o altro e che può essere utilizzato da un qualsiasi programma Basic 8.0; il Basic Paint è a nostro avviso una ragione sufficiente per comperare l'intero pacchetto.

Il programma può essere controllato sia da joystick che da mouse (il 1351 va benissimo). Le funzioni disponibili includono: zoom, cut and paste, erase, disegno a mano libera, linee, ecc.

Basic Paint vi permette di creare automaticamente delle scatole, dei cerchi, degli elissi o dei poligoni, i disegni possono essere integrati con vari tipi di font di varie dimensioni. Per disegnare è possibile servirsi anche di una griglia che viene sovraimpressa facilitando in questo modo la realizzazione del disegno.

Per poter utilizzare appieno le capacità del Basic 8.0 è richiesta della memoria di schermo addizionale. Dal momento che qualsiasi operazione da voi effettuata all'interno della macchina provoca l'interruzione della garanzia vi consigliamo di rivolgervi per una eventuale sostituzione del chip RAM al vostro rivenditore o a del personale specializzato per tali operazioni.

Il programma viene venduto al pubblico al prezzo di $39.95 ed è disponibile presso LAGO s.n.c. - Via Napoleona, 16 - 22100 Como - Tel. 031/300174

Accordo C.T.O. - Infogrames

Nel luglio 1988, la società francese INFOGRAMES, leader nel settore dei video-game per personal computer, e la società C.T.O. di Bologna, hanno firmato degli accordi per la distribuzione esclusiva dei prodotti Infogrames in Italia.

Una gamma di software prestigiosi, tra i quali le ultime novità ancora inedite in Italia, sarà presentata allo SMAU 1988 allo stand della COMMODORE ITALIA.

Per il momento, la distribuzione esclusiva riguarderà la gamma Commodore (C-64 e Amiga) e i compatibili PC; i prodotti presentati saranno i seguenti: CAPTAIN BLOOD, MACADAM BUMPER, AQUAMAN VERSUS THE YELLOW SHADOW, BOBO, I PASSEGGERI DEL VENTO, BUBBLE GHOST, WARLOCK'S QUEST, SPIDERTRONIC, CAD 3D, LAST MISSION, LIVINGSTONE.

Partner d'eccezione per prodotti eccezionali

Il gruppo editoriale INFOGRAMES è nato attraverso la fusione di tre società leader nel settore europeo dei videogame:

INFOGRAMES,
ERE INFORMATIQUE,
COBRA SOFT.

Il gruppo Infogrames è già presente in 15 paesi e realizza più del 30% della sua cifra d'affari con l'export. Tra le caratteristiche principali della sua attività ricordiamo una costante ricerca nella creatività dei video giochi, specie tramite l'abbinamento col mondo del fumetto, del cinema, dei cartoni animati e dello sport.

Un centinaio di collaboratori compongono la sua struttura per un fattura che nel 1987 ha raggiunto la cifra di circa 15 milioni di dollari.

Il gruppo C.T.O. è una società estremamente dinamica e in continua crescita con sede a Bologna ed è rivolta al mercato del software attraverso due comparti principali:

— I prodotti professionali del mercato MS/DOS per quanto concerne la produttività individuale,

— i prodotti «consumer».

La base dell'attività del C.T.O. consiste nella ricerca, nello studio e nella commercializzazione di prodotti per l'informatica, direttamente importati e selezionati fra quelli delle migliori case produttrici mondiali.

I prodotti INFOGRAMES vanno ad aggiungersi ad altri importanti marchi esclusivi per l'Italia, tra i quali citiamo:

— LOGISTIX,
— PORTEX,
— PRECISION SOFTWARE LTD,
— LIFETREE SOFTWARE LTD,
— ELECTRONIC ARTS,
— PROGRESSIVE PERIPHERALS & SOFTWARE.

Nonché l'affermata linea COMMODORE SOFTWARE by C.T.O.

La flessibilità della struttura del C.T.O. si traduce in scelte di mercato che consentano una risposta efficace e funzionale alla domanda del settore informatico.

La rete di vendita del C.T.O. copre la totalità del mercato nazionale.

Per ulteriori informazioni, contattare:

INFOGRAMES: Sabine Robert (Public Relations)

84 rue du 1er mars 1943

69628 Villeurbanne Cedex

tel: 00 33 78 03 78 46

telex: 305 551

fax: 00 33 78 03 18 40

C.T.O.: Maria Rosa Piovani - Agenzia Technostudi

via Mascarella, n. 57

40126 Bolgona

tel: (051) 241 436

Bob Morane Space Ace

Una storia nel XXV secolo... un punto nello spazio intersiderale...

Bob Morane si era immaginato che il satellite sarebbe loro apparso come un prodigioso gioiello nel vertiginoso astuccio di seta blu del vuoto interstellare. Tuttavia la realtà era completamente diversa perché ora che il congegno era davanti a loro, gli appariva piuttosto come un gigantesco ragno metallico sospeso nelle tenebre originali e pronto a divorare gli audaci che gli si avvicinano.

"Non molto simpatico, il meccanico" constata Bill Ballantine che legato al fianco del suo amico su una delle sedie dello scafo osservava anch'egli il satellite allo schermo.

Niente di molto simpatico, questo era certo, ma tuttavia i due uomini pensavano di non avere niente da temere da esso per il momento. Il satellite non era che una cosa smarrita, dopo lungo tempo liberata dalla morte e dall'oblio: era il 2500 e il satellite era stato messo in orbita nel XX secolo. Se ci fossero stati degli esseri umani all'interno, questi sarebbero dovuti essere morti da tempo.

"Io spero che l'accostamento riesca senza ostacoli" fece Bill Ballantine nel suo trasmettitore.

Bob Morane era immerso nei suoi pensieri. "Quella potenza aveva senz'altro avuto bisogno dell'Ombra Gialla per realizzare l'impresa d'inviare in orbita un satellite di quella taglia, nonostante le grandi Potenze dell'epoca ne fossero state incapaci!".

"Mi senti?" riprese più forte Bill Ballantine.

"Spero che vada tutto bene" rispose Bob Marane e cominciò la manovra di avvicinamento. Nel momento in cui il suo indice toccò i comandi, si sprigionò un grande raggio rosso ed egli sentì un'enorme scossa. I due viaggiatori ebbero subito l'impressione di essere annientati. Poi ebbero un'oscillazione nauseante attraverso l'infinito. Quando ripresero coscienza, il satellite sembrava essere invecchiato...

Subito Bob Morane lanciò un grido. "Guarda il tempometro" disse a Bill con voce fioca. Subito il colosso balzò e ruggì "10000! 10000 dopo Cristo, ma non è possibile".

Ma il tempometro non poteva sbagliare, erano stati proiettati attraverso il tempo, senza sapere come, sino al decimo millennio dopo Cristo...

Così incominciava il racconto d'avventura Space Ace... (scoprirete il seguito nel libro "Il satellite dell'ombra gialla").

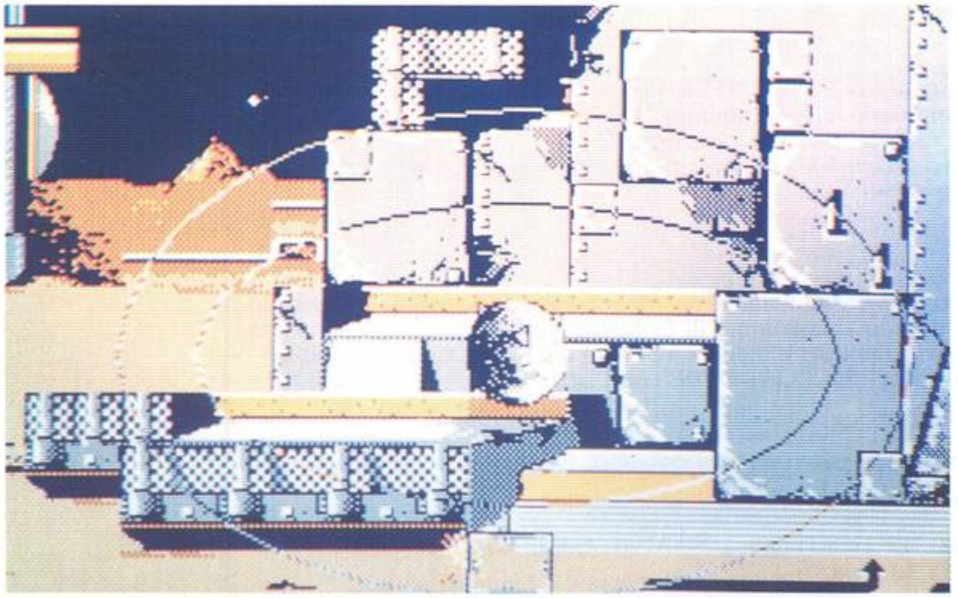
Un libro tascabile...

Nel libro tascabile che accompagnava l'edizione originale di Bob Morane Space Ace si trovava un romanzo (Il satellite dell'ombra gialla), un BD Tharn Il principe delle stelle, una guida (per sapere l'essenziale della conquista spaziale, dell'addestramento...), una avventura di cui voi sareste stati gli eroi inediti (Il satellite malefico), dove avreste dovuto far fallire il progetto dell'Ombra Gialla: la stessa che aveva messo in orbita il satellite mortale!

Purtroppo, a causa dello spazio disponibile vi dovrete accontentare del solo gioco.

Il software...

Bob Morane Space Ace è un programma in cui voi dirigerete l'eroe attraverso lo spazio intersiderale. La sua missione: ritrovare il suo fedele amico scozzese Bill Ballantine rapito dalla terribile Ombra Gialla e fatto prigioniero in un immenso satellite prigione. Dovrete liberarlo diffidando dei numerosi robot androidi del vostro nemico e dei falsi Bill Ballantine: dei replicanti che ha



fatto costruire l'Ombra Gialla per poter meglio indurvi in errore. Affrontare le trappole strane e terribili del continuo spazio-tempo non è una cosa semplice!

Vi sposterete nello spazio attraverso molteplici stanze e labirinti infestati da robot distruttori, da trappole mortali e da replicanti. A vostra disposizione delle pistole laser, dei raggi paralizzanti... Una vera panoplia dello spazio. L'insieme delle azioni sono controllate molto semplicemente dalla tastiera dove leve di comando vi permettono di concentrarvi efficacemente sul vostro compito difficile. Dall'inizio le vostre chances di riuscire sono vicine allo 0%. Smentite questa probabilità e provate, con Bob Morane, ad essere più forti e più astuti!...

Uso del programma

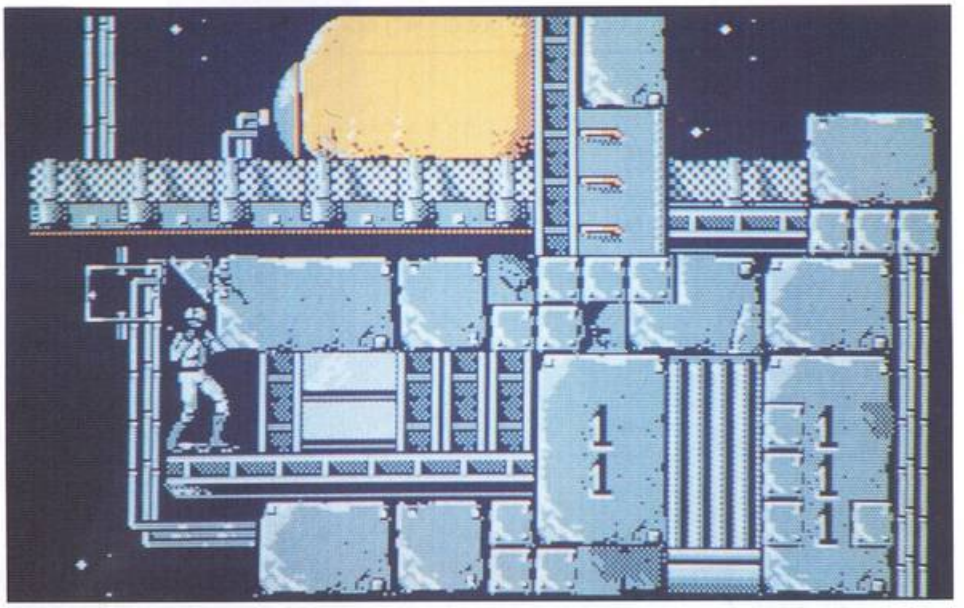
Versione in cassetta:
— Premere il tasto Play del registratore.
— Se si usa un C 128, digitare GO 64 e confermare premendo il tasto Y e il Return.
— Poi, anche con il C 64, premere contemporaneamente i tasti Shift e Run-/Stop.
— Il programma verrà caricato automaticamente.

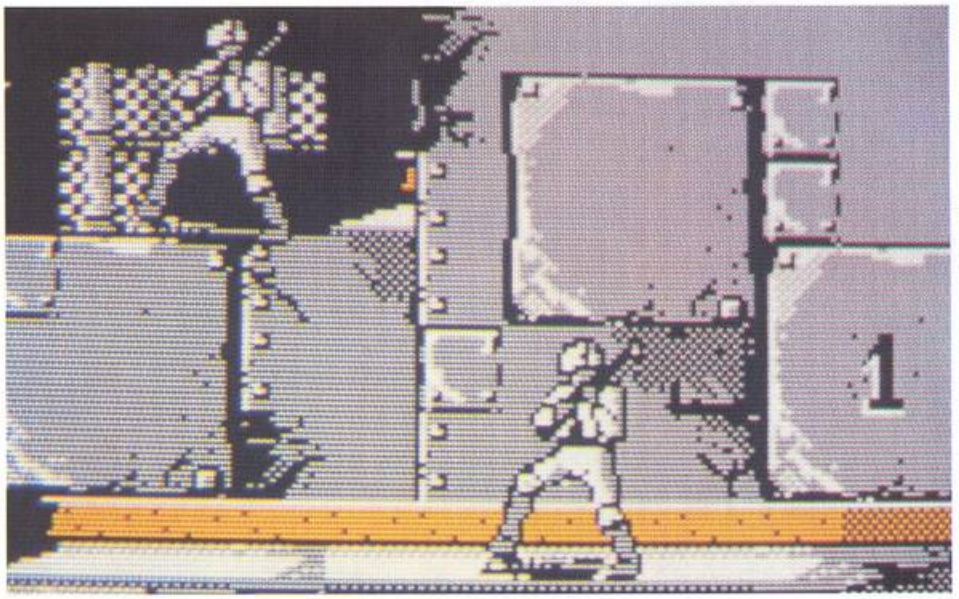
Versione su dischetto:
— Inserire il dichetto nel drive.
— Se si usa un C 128, digitare GO 64 e confermare premendo il tasto Y e il Return.
— Poi, anche con il C 64, digitare LOAD"?**",8,1.
— Premere il tasto Return.

Attenzione: il gioco si trova sul lato B del dischetto.

Principi del gioco

Ci si deve sbarazzare di tutti i nemici spediti dall'Ombra Gialla per distruggerci. Sullo schermo si può vedere il mirino del proprio fucile laser. Questo





è molto soffisticato perche ci indica dove si trovano i nostri nemici. Emette anche un segnale acustico che diviene sempre più alto all'avvicinarsi di un nemico pronto a colpire.

Comandi

Per muovere il mirino, spostare il joystick nella direzione desiderata.

Per colpire, premere il tasto del fuoco.

Per proteggersi con un campo magnetico premere la barra spaziatrice.

Questo tende gradualmente a perdere la propria efficacia, è pertanto consigliabile usarlo solo nei casi di estrema necessità.

Per fermare il gioco, e per poi rifarlo ripartire, premere il tasto STOP.

Vi accorgerete rapidamente che la Yellow Shadow vi ha attirato in una trappola... dovrete essere molto astuti e attenti per potervi mantenere in vita a lungo...

Errata corrige

Nel listato del programma FAST.LISTER 128 pubblicato sul numero 10, sono state omesse le ultime linee di programma: (12040-14390), contenenti i DATA relativi alle stringhe di decodifica dei caratteri.

Sono presenti inoltre due errori di battitura nelle linee data 11003 e 12005, che è possibile correggere come segue:

11003 DATA BUMP
12005 DATA TEMPO

I due errori succitati sono presenti anche nel file programma compilato, per cui chi ha ordinato il dischetto con i programmi di COMMODORE PROFESSIONAL dovrà eseguire le seguenti operazioni per eliminare gli errori dal programma compilato:

1) Caricare in memoria il file FLIST.CODE con l'istruzione

DLOAD "FLIST.CODE" <RETURN>

2) Digitare le seguenti istruzioni:

GRAPHIC CLR:BANK 0 <RETURN>

POKE DEC("2FBA"),DEC("4D") <RETURN>
POKE DEC("2FBB"),DEC("50") <RETURN>
POKE DEC("3000"),DEC("4D") <RETURN>
POKE DEC("3001"),DEC("50") <RETURN>

3) Cancellare il file "FLIST.CODE" con l'istruzione:

SCRATCH "FLIST.CODE" <RETURN>

4) Salvare il file corretto con

DSAVE "FLIST.CODE" <RETURN>

```
12040 :
12041 rem stringhe di decodifica caratt
      eri tra doppi apici
12042 rem ------------- codifica parzia
      le ----------------
13000 data 1,ctrl<a>
13010 data 2,ctrl<b>
13020 data 3,ctrl<c>
13030 data 4,ctrl<d>
13040 data 5,ctrl<2>
13050 data 6,ctrl<f>
13060 data 7,ctrl<g>
13070 data 8,ctrl<h>
13080 data 9,ctrl<i>
13090 data 10,ctrl<j>
13100 data 11,ctrl<k>
13110 data 12,ctrl<l>
13120 data 13,ctrl<m>
13130 data 14,ctrl<n>
13140 data 15,ctrl<o>
13150 data 16,ctrl<p>
13160 data 17,down
13170 data 18,rvon
13180 data 19,home
13190 data 20,del
13200 data 21,ctrl<u>
13210 data 22,ctrl<v>
13220 data 23,ctrl<w>
13230 data 24,ctrl<x>
13240 data 25,ctrl<y>
13250 data 26,ctrl<z>
13260 data 27,esc
13270 data 28,ctrl<3>
13280 data 29,left
13290 data 30,ctrl<6>
13300 data 31,ctrl<7>
13310 data 32,spc
13320 data 129,cmdr<1>
13330 data 133,f1
13340 data 134,f3
13350 data 135,f5
13360 data 136,f7
13370 data 137,f2
13380 data 138,f4
13390 data 139,f6
13400 data 140,f8
13410 data 144,ctrl<1>
13420 data 145,up
13430 data 146,rvoff
13440 data 147,clr
13450 data 148,inst
13460 data 149,cmdr<2>
13470 data 150,cmdr<3>
13480 data 151,cmdr<4>
13490 data 152,cmdr<5>
13500 data 153,cmdr<6>
13510 data 154,cmdr<7>
13520 data 155,cmdr<8>
13530 data 156,ctrl<5>
13540 data 157,right
13550 data 158,ctrl<8>
13560 data 159,ctrl<4>
13570 data 160,shift<spc>
13580 data -1,
13590 :
13600 rem stringhe di codifica tasti co
      mmodore/key e tasti particolari
13610 rem -------------------- codifica
```

```
      totale ----------------------
14000 data 171,cmdr<q>
14010 data 179,cmdr<w>
14020 data 177,cmdr<e>
14030 data 178,cmdr<r>
14040 data 163,cmdr<t>
14050 data 183,cmdr<y>
14060 data 184,cmdr<u>
14070 data 162,cmdr<i>
14080 data 185,cmdr<o>
14090 data 175,cmdr<p>
14100 data 164,cmdr<'>
14110 data 186,shift<'>
14120 data 223,cmdr<*>
14130 data 192,shift<*>
14140 data 166,cmdr<+>
14150 data 219,shift<+>
14160 data 220,cmdr<->
14170 data 221,shift<->
14180 data  92,pound:       rem  <\>  s
      imbolo lira
14190 data 168,cmdr<pound>: rem <"("> c
      ommodore lira
14200 data 169,shift<pound>:rem <")"> s
      hift lira
14210 data  94,uparrow:     rem  <^>  f
      reccia in su
14220 data  95,backarrow:   rem  <_>  f
      reccia a destra
14230 data 176,cmdr<a>
14240 data 174,cmdr<s>
14250 data 172,cmdr<d>
14260 data 187,cmdr<f>
14270 data 165,cmdr<g>
14280 data 180,cmdr<h>
14290 data 181,cmdr<j>
14300 data 161,cmdr<k>
14310 data 182,cmdr<l>
14320 data 173,cmdr<z>
14330 data 189,cmdr<x>
14340 data 188,cmdr<c>
14350 data 190,cmdr<v>
14360 data 191,cmdr<b>
14370 data 170,cmdr<n>
14380 data 167,cmdr<m>
14382 data 255,shift<uparrow>:rem shift
      /freccia in su (pi greco)
14390 data -1,
```

Anticipazioni

C 128

- Portfolio: un programma che vi permette un'analisi sempre aggiornata dei vostri investimenti
- Smart shopper: come diventare degli acquirenti smaliziati
- Colori in Hi-res: grafica Hi-res in ottanta colonne con i 16 colori del C 128

C 64

- 64 dos shell: un'utility versatile ed utilissima per la gestione dei dischetti
- Un C 64 e tanti dati: gestire molte informazioni con un piccolo computer può essere difficile... se non si conoscono i trucchi del mestiere (seconda parte)
- Driver 1541: una routine per leggere i dischetti blocco per blocco
- Begli programmi: alcuni utili consigli per rendere più leggibili, eleganti e facili da correggere i vostri programmi
- Il compressore: come far stare un numero elevato di disegni in hi-res sui vostri dischetti occupando la metà dello spazio necessario
- Hi Res text: un programma che vi permetterà di modificare in modo facile e interessante i soliti caratteri del vostro C 64

LISTINO LIBRI JACKSON

CODICE	TITOLO	PREZZO	
	INFORMATICA: CONCETTI GENERALI		
511 A	COME PROGRAMMARE	15.000	
503 A	PROGRAMMAZIONE STRUTTURATA, CORSO DI AUTOISTRUZIONE	15.000	
101 H	TERMINI DELL'INFORMATICA E DELLE DISCIPLINE CONNESSE	50.000	
539 A	LOGICA E DIAGRAMMI A BLOCCHI: TECNICHE DI PROGRAMMAZIONE	40.000	
526 P	DATA BASE: CONCETTI E DISEGNO	22.500	
GYS190	TRADUTTORI DI LINGUAGGI	26.000	
G 240	PAROLE BASE DELL'INFORMATICA	8.000	
GYS245	CONCETTI DI INFORMATICA	43.000	
GYS248	DATA PROCESSING	45.000	
GY 264	DATA FILE	50.000	
GYS266	ARCHITETTURE DI SISTEMA	32.000	
GY 354	SISTEMI INTELLIGENTI	28.000	
CZ 419	ANALISI E PROGRAMMAZIONE	11.000	
158 EC	INFORMATICA DI BASE I CONCETTI FONDAMENTALI HARDWARE E SOFTWARE	55.000	
526 A	VOI E L'INFORMATICA	15.000	
100 H	DIZIONARIO DI INFORMATICA	59.000	
GY 551	I LINGUAGGI DELLA 4ª GENERAZIONE	65.000	
GYS552	PRIMA DEL LINGUAGGIO LA PROGRAMMAZIONE	35.000	
GYS 559	C.S.P. - PROCESSI SEQUENZIALI	49.000	
GYS 546	ALGORITMI FONDAMENTALI	54.000	
GY 618	SISTEMI ESPERTI	28.000	
047 T	MICROPROCESSORI	14.500	
048 T	DATA BASE	14.500	
049 T	FILE	14.500	
CI 686	CAPIRE IL PERSONAL COMPUTER	35.000	
G 540	MODELLI MATEMATICI E SIMULAZIONE	56.000	
GE 688	ENCICLOPEDIA MONOGRAFICA DI ELETTR. E INF. VOLUME I	58.000	
GE 689	ENCICLOPEDIA MONOGRAFICA DI ELETTR. E INF. VOLUME II	58.000	
GY 629	SOFTWARE DI BASE - Strumenti di sviluppo	52.000	
	INFORMATICA: SISTEMI OPERATIVI		
G 223	UNIX LA GRANDE GUIDA	70.000	
GY 272	SISTEMI OPERATIVI PER MICROCOMPUTER	25.000	
GY 273	MS-DOS LA GRANDE GUIDA	45.000	
510 P	CP/M CON MP/M	29.000	
CZ 538	MS DOS 2 E 3	49.000	
G 543	XENIX	45.000	
R 588	LAVORARE CON XENIX	70.000	
GYS271	SISTEMI OPERATIVI	55.000	
R 615	I COMANDI DI XENIX MAIL	12.500	
092 D	SOFTWARE DI BASE E SISTEMI OPERATIVI	7.000	
093 D	CP/M IL "SOFTWARE BUS"	7.000	
094 D	MS-DOS E PC-DOS LO STANDARD IBM	7.000	
009 H	UNIX	8.500	
011 H	CP/M	8.500	
044 T	MS DOS	14.500	
045 T	PC DOS	14.500	
R 628	MICROSOFT OS/2	50.000	
046 T	UNIX	14.500	
MS 02 E	COFANETTO "MS-DOS" 5"¼ - Corso autoistruzione	156.000	F
R 600	MS DOS ADVANCED - Il Manuale del Programmatore	55.000	
GY 663	UNIX PROGRAMMAZIONE AVANZATA	55.000	
BY 724	GUIDA AI SISTEMI OPERATIVI	29.000	
BY 744	UNIX: CONCETTI, STRUTTURE, UTILIZZO	43.000	
R 761	MS-DOS REFERENCE GUIDE	14.500	
	INFORMATICA: LINGUAGGI		
501 A	IMPARIAMO IL PASCAL	16.000	
502 A	INTRODUZIONE AL BASIC	25.000	
500 P	PASCAL MANUALE E STANDARD DEL LINGUAGGIO	16.000	
329 A	PROGRAMMARE IN ASSEMBLER	14.000	
513 A	PROGRAMMARE IN BASIC	8.000	
514 A	PROGRAMMARE IN PASCAL	19.000	
516 A	INTRODUZIONE AL PASCAL	39.000	
517 P	DAL FORTRAN IV AL FORTRAN 77 (II ED.)	32.000	
521 A	50 ESERCIZI IN BASIC	17.000	
525 A	BASIC PER TUTTI	23.000	

CODICE	TITOLO	PREZZO	
534 A	MANUALE DEL BASIC	45.000	
509 A	LOGO: POTENZA E SEMPLICITÀ	20.500	
507 B	TUO PRIMO PROGRAMMA IN BASIC (II)	19.500	
533 A	BASIC DALLA A ALLA Z	19.000	
540 A	LINGUAGGIO ADA	19.500	
541 P	LINGUAGGIO C	25.000	
542 P	COBOL STRUTTURATO: CORSO DI AUTOISTRUZIONE	50.000	
508 P	PROGRAMMARE IN C	39.000	
G 233	COBOL PER MICROCOMPUTER	35.000	
GYS246	ESERCIZI DI FORTRAN	20.000	
GYS247	ESERCIZI IN PASCAL: ANALISI DEI PROBLEMI	29.000	
GYS254	PROGRAMMAZIONE IN LINGUAGGIO ADA	42.000	
GY 270	APL PER IL P.C. IBM	25.000	
GYS274	DAL PASCAL AL MODULA 2	26.000	
GYS311	LINGUAGGIO C IL LIBRO DELLE SOLUZIONI	24.000	
GYS328	APPLICAZIONI IN PASCAL	32.000	
GY 535	TURBO PASCAL	29.000	
G 544	"C" LIBRARY	49.000	
GYS550	PROLOG - LINGUAGGIO E APPLICAZIONE	32.000	
R 589	TURBOPASCAL - LIBRERIA DI PROGRAMMI	45.000	
042 T	LINGUAGGIO C	12.500	
108 D	FORTH ANATOMIA DI UN LINGUAGGIO	7.000	
107 D	FORTRAN E COBOL LINGUAGGI SEMPRE VERDI	7.000	
086 D	ED È SUBITO BASIC VOL. 1	7.000	
087 D	ED È SUBITO BASIC VOL. 2	7.000	
034 T	PROLOG	14.000	
035 T	LISP	12.500	
001 H	COBOL	8.500	
006 H	PASCAL	8.500	
007 H	BASIC	8.500	
010 H	FORTRAN 77	8.500	
020 H	LOGO	8.500	
022 H	FORTH	8.500	
R 612	TURBO PROLOG	50.000	
GY 626	IL MANUALE DEL PASCAL	42.000	
GY 616	DEBUGGING C	55.000	
GY 687	DALLA PROGRAMMAZIONE STRUTTURATA AL PASCAL	42.000	
GY 634	FONDAMENTI DI COMMON LISP	40.000	
	INFORMATICA: LAVORO E SOCIETÀ		
519 P	COMPUTER GRAFICA	29.000	
800 P	ODISSEA INFORMATICA	50.000	
407 H	APPLICAZIONI DEL COMPUTER NELL'UFFICIO MODERNO	23.000	
802 H	INFORMATICA MUSICALE	27.000	
802 P	COMPUTERGRAPHIA	40.000	
806 P	COMPUTER PER L'INGEGNERIA EDILE	22.000	
807 P	COMPUTER PER IL MEDICO	19.000	
CI 231	COMPUTER IMAGE	40.000	
CI 241	ODISSEA INFORMATICA STRATEGIE CULTURALI PER UNA SOCIETÀ INF.	32.000	
G 400	COMPUTER GRAPHICS E ARCHITETTURA	27.000	
PV 409	COMPUTER GRAPHICS E MEDICINA	18.000	
GY 487	MEDICO & COMPUTER	45.000	
GY 548	INFORMATICA MEDICA	65.000	
PA 685	OFFICE AUTOMATION	28.000	
RA 596	DESKTOP PUBLISHING	35.000	
050 T	WORD	14.500	
	INFORMATICA: SOFTWARE PACCHETTI APPLICATIVI		
570 P	CONTABILITÀ COL PERSONAL COMPUTER	27.000	
525 P	WORDSTAR	24.000	
546 P	MANUALE DEL DBASE II	24.000	
578 P	PC NELL'ORG. DELLE PICCOLE AZIENDE: APPL. DEL MULTIPLAN	29.000	F
PP 219	LOTUS 1-2-3: GUIDA ITALIANA ALL'USO	21.000	
G 234	RIORDINO E GESTIONE DEGLI ARCHIVI APPLICAZIONI CON PFS-FILE	30.000	
PP 255	DBASE III GUIDA ITALIANA ALL'USO	45.000	
PA 282	MODELLI DECISIONALI PER IL MANAGER	50.000	
PA 288	PIANIFICAZIONE AZIENDALE PLANNING, MARKETING STRAT., BUDGETING	35.000	F
PP 310	LA GRANDE GUIDA LOTUS A SYMPHONY	70.000	
PP 326	MULTIPLAN CORSO D'ISTRUZIONE	40.000	F

CODICE	TITOLO	PREZZO	
PP 344	FRAMEWORK II - GUIDA ITALIANA ALL'USO	27.000	
PP 351	WORD PROCESSING	27.000	
PP 467	IMPARA 1-2-3 CON LA GRANDE GUIDA LOTUS	45.000	
PP 468	CHART - CORSO ISTRUZIONE	45.000	F
PP 473	IL NUOVO 1-2-3 GUIDA ALL'USO DELLA VERSIONE ITALIANA 2 LOTUS 1-2-3	29.000	
PA 474	BILANCIO, BUDGET, CASH FLOW	40.000	F
PP 475	DBASE III - CORSO DI PROGRAMMAZIONE	23.000	
PA 476	PREVISIONE, PIANIFICAZIONE, SIMULAZIONE CON LOTUS 1-2-3	60.000	F
PV 477	GUIDA ALLA BUSINESS GRAPHIC	20.000	
PP 480	AUTOCAD	40.000	
PP 481	RBASE 5000 - GUIDA ITALIANA ALL'USO	20.000	
PP 537	IL MANUALE DI WINDOWS	60.000	
PP 539	DBASE III - TECNICHE AVANZATE DI PROGRAMMAZIONE	42.000	
PP 545	APPLICAZIONI DI DBASE III	50.000	F
PA 566	MODELLI DECISIONALI CON LOTUS 1-2-3	40.000	F
PP 577	MANUALE DBASE III PLUS	49.000	
039 T	WORDSTAR	12.500	
040 T	LOTUS 1-2-3	12.500	
043 T	WINDOWS	12.500	
PP 621	I COMANDI DI DBASE III PLUS	12.500	
095 D	GUIDA AI PACKAGE APPLICATIVI MERCEOLOGIA DEL SOFTWARE	7.000	
096 D	VISICALC GUIDA RAPIDA ALL'UTILIZZO	7.000	
098 D	WORD PROCESSING	7.000	
103 D	LOTUS 1-2-3 E SIMPHONY IL FASCINO DELL'INTEGRAZIONE	7.000	
104 D	DBASE II E III I PRINCIPI DI DATABASE	7.000	
106 D	MULTIPLAN SPREADSHEET MULTISTRATO	7.000	
110 D	PACKAGE A CONFRONTO PROVE DEI SOFTWARE PIÙ DIFFUSI	7.000	
031 T	FRAMEWORK E FRAMEWORK II	12.500	
033 T	MULTIPLAN 2.02	12.500	
036 T	SYMPHONY	12.500	
038 T	REFLEX	12.500	
027 H	EASY SCRIPT	8.500	
033 H	PAGE MAKER	8.500	
034 H	PROJECT	8.500	
035 H	RBASE	8.500	
PP 611	GUIDA ALL'USO PROFESSIONALE REFLEX	55.000	
PP 636	MANUALE DI WORD	70.000	
PP 594	GUIDA ALL'USO PROFESSIONALE DI LOTUS 1-2-3	50.000	
PP 593	VENTURA - Il grande manuale	55.000	
R 671	LINGUAGGIO C - Reference guide	12.500	
051 T	I COMANDI DI LOTUS 1-2-3 - Reference guide	12.500	
PP 581	PROGRAMMARE IN FRED	40.000	
PP 631	dBASE III PLUS - Guida uso professionale	65.000	
PP 694	PROGRAMMARE IN WINDOWS	70.000	
PA 592	GESTIONE DELLA PRODUZIONE	40.000	F
PP 727	VENTURA - REFERENCE GUIDE	14.500	
PP 700	MATEMATICA CON LOTUS 1-2-3	35.000	F
R 574	MANUALE DELLE STAMPANTI LASER	25.000	
PP 641	AUTOCAD - Il grande manuale	55.000	
PP 728	VENTURA - Fogli stile	42.000	F
PP 741	WORD 3 e 4	59.000	F
PP 642	AUTOCAD Programmazione avanzata	65.000	F
BY 707	ORACLE	75.000	
PA 771	MODELLI PER LOTUS 1-2-3	28.000	F
	PERSONAL COMPUTER		
550 D	PROGRAMMI PRATICI IN BASIC	15.000	
515 H	BASIC E LA GESTIONE DEI FILE VOL. I: METODI PRATICI	15.000	
551 D	75 PROGRAMMI IN BASIC PER IL VOSTRO COMPUTER	12.000	
552 D	PROGRAMMI DI MATEMATICA E STATISTICA IN BASIC	20.000	
554 P	PROGRAMMI SCIENTIFICI IN PASCAL	29.000	
516 H	BASIC E LA GESTIONE DEI FILE - VOL. 2	17.000	
CH 182	COMPUTER HARDWARE REALIZZ. PRATICHE PER GLI HC PIÙ DIFFUSI	18.000	
CI 187	COMPUTER L'HOBBY E IL LAVORO	12.000	
G 235	GRAFICA PER PERSONAL COMPUTER	39.000	
GE 263	METODI DI INTERFACC. PERIFERICHE	43.000	
GE 402	CORSO DI AUTOISTRUZIONE PER MICROCOMPUTER VOL. 1 + VOL. 2	35.000	
PA 406	COME GESTIRE LA PICCOLA AZIENDA CON IL P.C.	22.000	
PP 408	BUSINESS IN BASIC	23.000	
CI 412	IL COMPUTER È UNA COSA SEMPLICE	15.000	

CODICE	TITOLO	PREZZO	
CC 415	CONTROLLO DEI DISPOSITIVI DOMESTICI CON IL P.C.	23.000	
159 GC	PERSONAL COMPUTER DAL SOFTWARE DI BASE ALLE APPLICAZIONI D'UFFICIO	55.000	
R 587	HARD DISK - LA GRANDE GUIDA	75.000	
084 D	INTRODUZIONE AI PERSONAL COMPUTER VIVERE COL PC	7.000	
099 D	SCRIVERE UN'AVVENTURA, 1000 AVVENTURE COL PROPRIO PC	7.000	
100 D	GRAFICA E BASIC LE BASI DELLA COMPUTERGRAFICA	7.000	
085 D	HARDWARE DI UN PERSONAL COMPUTER DENTRO E FUORI LA SCATOLA	7.000	
101 D	GESTIONE DEI FILE IN BASIC E PASCAL VOL. 1	7.000	
102 D	GESTIONE DEI FILE IN BASIC E PASCAL VOL. 2	7.000	
113 D	DISEGNARE COL PERSONAL COMPUTER	7.000	
105 D	PERSONAL E HOME COMPUTER A CONFRONTO	7.000	
112 D	SUONO E MUSICA COL PERSONAL COMPUTER	7.000	
109 D	COSTRUIRSI UN PERSONAL DATABASE	7.000	
097 D	GUIDA ALL'ACQUISTO DI UN PERSONAL COMPUTER	7.000	
088 D	TO DO OR NOT TO DO COME AVER CURA DEL PROPRIO PC	7.000	
089 D	SOFTWARE STRUTTURATO CON ELEMENTI DI PASCAL	7.000	
090 D	DIZIONARIO DI INFORMATICA	7.000	
091 D	BASI DELLA PROGRAMMAZIONE STENDERE UN PROG. COME SI DEVE	7.000	
004 H	PROGRAMMAZIONE	8.500	
015 H	PROGRAMMI DI STATISTICA	8.500	
	PERSONAL COMPUTER: COMMODORE		
347 D	VOI E IL VOSTRO COMMODORE 64	24.000	
348 D	COMMODORE 64 - IL BASIC	28.000	
400 D	FACILE GUIDA AL COMMODORE 64	13.500	
400 B	COMMODORE 64 - FILE	19.000	
409 B	COMMODORE 64 - LA GRAFICA E IL SUONO	34.000	C
570 D	MATEMATICA E COMMODORE 64	26.500	C
350 D	LIBRO DEI GIOCHI DEL COMMODORE 64	24.000	C
575 D	TECNICHE DI PROGRAMMAZIONE SUL COMMODORE 64	16.500	
572 D	LINGUAGGIO MACCHINA DEL COMMODORE 64	35.000	F
576 D	SISTEMA TOTOMAC: LA NUOVA FRONTIERA DEL TOTOCALCIO	29.000	C
548 B	64 PERSONAL COMPUTER E C64	45.000	
SDP222	STATISTICA AD UNA DIMENSIONE CON IL C64	24.000	
CC 260	AVVENTURE (COMMODORE 64)	20.000	C
CC 320	AMIGA HANDBOOK	35.000	
CC 322	COMMODORE 128 OLTRE IL MANUALE	29.000	
CC 323	PROGRAMMI PER COMMODORE 128	29.000	C
CZ 541	128 E 64 - LE PERIFERICHE	32.000	
CC 564	MANUALE RIPARAZIONE C64	55.000	
CZ 532	MANUALE DI AMIGA	39.000	
002 H	COMMODORE 64	8.500	
CC 658	GRAFICA E SUONO PER C64 - 64PC - C128	35.000	F
CC 657	MANUALE DEL COMMODORE C64 - C64PC - C128	35.000	F
CC 627	AMIGA 500	55.000	
CC 750	C.128 LA GRANDE GUIDA	50.000	
CC 749	C.64 LA GRANDE GUIDA	50.000	
	PETER NORTON		
R 734	MANUALE DEL DOS	55.000	
R 736	INSIDE PC IBM	63.000	F
R 733	HARD DISK COMPANION	60.000	
R 735	LINGUAGGIO ASSEMBLY PER PC IBM	72.000	2F
	PERSONAL COMPUTER: IBM		
564 D	PROGRAMMI UTILI PER IBM PC	19.000	
G 217	GRAFICA PER IL PERSONAL COMPUTER IBM	39.000	
GY 319	PC IBM MANUALE DEL LINGUAGGIO MACCHINA	45.000	
GY 335	MAPPING PC IBM GESTIONE DELLA MEMORIA	42.000	
PP 407	MANUALE BASE DEL PC IBM	22.000	
041 T	PC IBM	12.500	
R 609	SOLUZIONI AVANZATE PER IL PROGRAMMATORE	60.000	
CZ 751	AVVENTURE PER MS-DOS	35.000	F
RA 484	GUIDA ALLE RETI DI PC IBM	46.000	

CODICE	TITOLO	PREZZO	
	PERSONAL COMPUTER: OLIVETTI		
401 P	PRIMO LIBRO PER M24: MS DOS E GW BASIC	28.000	
401 B	OLIVETTI M10: GUIDA ALL'USO	18.000	
CL 216	BASIC IN 30 ORE PER M24 ED M20	32.000	
CZ 483	MANUALE OLIVETTI M19	42.000	
CZ 536	MANUALE PC 128 OLIVETTI PRODEST	29.000	
CZ 582	PROGR. PER PC 128 OLIVETTI PRODEST (CASS.)	27.000	
	PERSONAL COMPUTER: MSX		
CZ 181	30 PROGRAMMI PER MSX	20.000	C
417 D	MSX: IL BASIC	23.000	
CC 261	AVVENTURE (MSX)	20.000	C
CC 289	SUPER PROGRAMMI PER MSX	35.000	C
CC 336	MSX LA GRAFICA	25.000	
111 D	STANDARD MSX	7.000	
	PERSONAL COMPUTER: APPLE		
331 P	APPLE II GUIDA ALL'USO	31.000	
416 P	MACINTOSH NEGLI AFFARI: MULTIPLAN E CHART	16.500	
424 P	UN MAC PER AMICO: USO, APPLICAZIONI E PROGRAMMI PER MACINTOSH	12.000	
PP 224	MACINTOSH ARTISTA: MACPAINT E MACDRAW	16.000	
CCP277	APPLE IIC GUIDA ALL'USO	45.000	
CC 312	PROGRAMMI PER APPLE IIC	13.000	
CC 417	PROGRAMMI COMM. E FINANZIARI CON APPLE	22.000	
340 H	APPLE MEMO	15.000	
CC 576	IL MANUALE DELL'APPLE II GS	28.000	
003 H	APPLE IIE IIC	8.500	
CC 665	MICROSOFT BASIC PER APPLE MACINTOSH	32.000	F
	PERSONAL COMPUTER: ATARI - AMSTRAD - SHARP		
540 H	BASIC ATARI	18.000	
CC 330	PROGRAMMI PER AMSTRAD CPC 464 CPC 664 - CPC 6128	29.000	C
CC 331	PROGRAMMI PER ATARI 130XE	19.000	C
CC 471	MANUALE ATARI 520 ST E 1040 ST	28.000	
CC 486	WORD PROCESSING CON AMSTRAD PCW 8256/8512	35.000	
032 T	AMSTRAD PCW 8256 e PCW 8512	14.000	
028 H	AMSTRAD 464 E 664	8.500	
	COMMUNICATION E TELEMATICA		
309 A	PRINCIPI E TECNICHE DI ELABORAZIONE DATI	20.000	
518 D	TELEMATICA	28.000	
528 P	TRASMISSIONE DATI	27.000	
617 P	RETI DATI: CARATTERISTICHE, PROGETTO E SERVIZI TELEMATICI	40.000	
GYS314	ELABORAZIONE DIGITALE DEI SEGNALI: TEORIA E PRATICA	25.000	
PA 327	BANCHE DATI RICERCA ONLINE	26.000	
158 LC	COMUNICAZIONI DALLE ONDE ELETTROMAGNETICHE ALLA TELEMATICA	55.000	
CC 472	MODEM E PC USO E APPLICAZIONI	25.000	
GTS478	RETI LOCALI	44.000	
GTS479	IL MODEM - TEORIA, FUNZIONAMENTO	28.000	
R 542	TRASMISSIONE DATI PER PC	31.000	
GT 555	LA TELEMATICA NELL'UFFICIO	35.000	
R 601	COLLEGAMENTO TRA MICRO E MAINFRAME	39.000	
BT 655	MANUALE DI TV E VIDEO COMMUNICATION	45.000	
	ELETTRONICA DI BASE E TECNOLOGIA		
201 A	CORSO DI ELETTRONICA FONDAMENTALE CON ESPERIMENTI	35.000	
204 A	ELETTRONICA INTEGRATA DIGITALE	50.000	
205 A	MANUALE PRATICO DI PROGETTAZIONE ELETTRONICA	35.000	
200 A	SISTEMI DIGITALI: MANUTENZIONE, RICERCA ED ELIMINAZIONE GUASTI	28.500	
GES262	TECNOLOGIE VLSI	70.000	
GES390	ELETTRONICA INTEGRATA DIGITALE IL LIBRO DELLE SOLUZIONI	17.000	
CE 411	LA FISICA DEI SEMICONDUTTORI	10.000	
158 PC	ELETTRONICA DI BASE I FONDAMENTI DELL'ELETTRONICA ANALOGICA	55.000	
158 CC	ELETTRONICA DIGITALE VOL. 1 DALLE PORTE LOGICHE AI CIRCUITI INTEGRATI	55.000	
158 DC	ELETTRONICA DIGITALE VOL. 2 DAI BUS AI GATE ARRAY	55.000	
158 GC	ELETTROTECNICA ELETTROSTATICA ELETTROMAGNETISMO RETI ELETTR.	55.000	

CODICE	TITOLO	PREZZO
	ELETTRONICA APPLICATA	
601 B	TIMER 555	10.000
203 A	INTRODUZIONE AI CIRCUITI INTEGRATI DIGITALI	10.000
612 P	MANUALE DEGLI SCR VOL. 1	28.000
613 P	MANUALE DI OPTOELETTRONICA	15.000
614 A	FIBRE OTTICHE	15.000
GE 403	JFET MOS E DATA BOOK	20.000
GE 404	TRANSISTOR DATA BOOK	32.000
GE 405	METODI DI PROTEZIONE CONTRO LE SOVRATENSIONI	17.000
CE 413	IL MANUALE DEGLI SCR E TRIAC	15.000
CE 421	MANUALE DEI FILTRI ATTIVI	29.000
CE 423	MANUALE DEI PLL PROGETTAZIONE DEI CIRCUITI	29.000
CE 425	MANUALE DEGLI AMPLIFICATORI OPERAZIONALI	29.000
CE 429	250 PROGETTI CON GLI AMPLIFICATORI DI NORTON	39.000
CE 431	MANUALE DEI CMOS	25.000
CE 485	IL COLLAUDO DELLE SCHEDE	18.000
BE 557	I TRASDUTTORI	43.000
BT 585	FIBRE OTTICHE	29.000
BE 578	MANUALE DI ELETTRONICA	29.000
BE 558	IL MANUALE DEL TECNICO ELETTRONICO	51.000
BE 610	GUIDA ALLA STRUMENTAZIONE ELETTRONICA	34.000
BE 619	MULTIMETRI DIGITALI	42.000
BE 639	ENCICLOPEDIA DEI CIRCUITI INTEGRATI	60.000
BE 654	MANUALE DI ELETTRONICA DEL COMPUTER	20.000
701 P	MANUALE PRATICO DEL RIPARATORE RADIO TV	29.000
705 P	IMPIEGO PRATICO DELL'OSCILLOSCOPIO	17.500
615 P	PROGETTAZIONE DI SISTEMI DI ALTOPARLANTI	21.000
CE 427	L'ELETTRONICA A STATO SOLIDO	25.000
BE 718	77 SCHEDE PER IL RIPARATORE TV	40.000
BE 723	MISURE DEI CIRCUITI ELETTRONICI	26.000
BE 721	MANUALE PRATICO DI ELETTRONICA DIGITALE	26.000
BE 684	IL MANUALE DEI CMOS	35.000
BE 731	IL MANUALE DEGLI AMPLIFICATORI OPERAZIONALI	39.000
	ELETTRONICA: MICROPROCESSORI	
310 P	NANOBOOK Z80 VOL. 1	20.000
007 A	BUGBOOK VII	17.000
314 P	TECNICHE DI INTERFACCIAMENTO DEI MICROPROCESSORI	31.000
312 P	NANOBOOK Z80 VOL. III	25.000
320 P	MICROPROCESSORI DAI CHIPS AI SISTEMI	29.000
324 P	PROGRAMMAZIONE DELLO Z80 E PROGETTAZIONE LOGICA	21.500
326 P	Z80 PROGRAMMAZIONE IN LINGUAGGIO ASSEMBLY	50.000
328 D	PROGRAMMAZIONE DELLO Z80	40.000
504 B	APPLICAZIONI DEL 6502	17.000
503 B	PROGRAMMAZIONE DEL 6502	35.000
505 B	GIOCHI CON IL 6502	19.500
G 220	8086-8088 PROGRAMMAZIONE	40.000
GY 265	ASSEMBLER PER IL 68000	70.000
CE 410	IMPIEGO DELLO Z80	23.000
158 HC	MICROPROCESSORI ARCHIT. PROGR. E INTERFAC. DEI MP DA 4 A 32 BIT	55.000
013 H	ASSEMBLER 6502	8.500
016 H	ASSEMBLER Z80	8.500
021 H	ASSEMBLER 68000	8.500
025 H	ASSEMBLER 8086-8088	8.500
029 H	ASSEMBLER 80286	8.500
GE 567	80286 ARCHITETTURA E PROGRAMMAZIONE	58.000
GY 603	80386 ARCHITETTURA E PROGRAMMAZIONE	37.000
	AUTOMAZIONE	
208 A	CONTROLLORI PROGRAMMABILI	24.000
616 P	CONTROLLO AUTOMATICO DEI SISTEMI	29.500
GES251	STRUTTURA E FUNZIONAMENTO DEI CONTROLLI NUMERICI	29.000
GES252	CONTROLLI NUMERICI: PROGRAMMAZIONE E APPLICAZIONI	28.000
G 399	30 APPLICAZIONI DI CAD	29.000
G 401	CAD/CAM & ROBOTICA	28.000
CI 414	DAL CHIP ALLA ROBOTICA	15.000
GE 547	LA PROGETTAZIONE AUTOMATICA	32.000
GE 564	ROBOTICA - Fondamenti e applicazioni	38.000

CODICE	TITOLO	PREZZO	
	DIZIONARI ENCICLOPEDICI		
DS 498	FISICA	14.000	
DS 499	MATEMATICA	14.000	
DS 522	GEOLOGIA	14.000	
DS 524	ELETTRONICA	14.000	
DS 525	ASTRONOMIA	14.000	
DS 526	CHIMICA	14.000	
DS 527	RAGIONERIA GENERALE	14.000	
DS 528	RAGIONERIA APPLICATA	14.000	
DS 529	BIOLOGIA	14.000	
DS 530	MECCANICA	14.000	
DS 531	INFORMATICA	14.000	
	ARGOMENTI VARI		
704 D	MANUALE PRATICO DI REGISTRAZIONE	10.000	
706 A	COMUNICAZIONI RADIO IN MARE	18.000	
800 H	FENDER, STORIA DI UN MITO	28.000	
CZ 748	TOTOCALCIO, ENALOTTO, TOTIP	35.000	F
	SOFTWARE E MANAGEMENT TOOLS		
CZ 469	GRAFIX - DISEGNARE CON IL PC	50.000	F
TP 606	CORSO AUTOISTRUZIONE LOTUS 1-2-3 (VERS. ITALIANA) F - MS DOS	90.000	F
TY 605	CORSO AUTOISTRUZIONE SUL SISTEMA MS DOS - FLOPPY	50.000	F
TY 640	TURBO PASCAL - LIBRERIA DI PROGRAMMI	40.000	F
TP 643	CORSO AUTOISTRUZIONE LOTUS 1-2-3 (INGLESE)	90.000	F
TP 608	BUDGET STRATEGICO (LOTUS 1-2-3)	100.000	F
TP 614	GESTIONE DELLE COMMESSE DI PRODUZIONE	100.000	F
TP 623	CONTROLLO DELLE VENDITE (CON MULTIPLAN)	100.000	F
TP 625	GESTIONE DEL PERSONALE (LOTUS 1-2-3)	100.000	F
TP 677	GESTIONE DELLE COMMESSE CON MULTIPLAN 2.0	100.000	F
TP 673	PREVENTIVO E CONSUNTIVO DEI COSTI - CON LOTUS 1-2-3 VERS. 2 E MULTIPLAN 2.0	100.000	F
TP 660	1-2-3 LIBRERIA DI MACRO	60.000	F
TY 691	SUPER SCREEN - UTILITY PER I PROGRAMMATORI	50.000	F
TY 690	PC DOCTOR UTILITY - RECOVERING DEI FILE	60.000	F
TP 644	STATISTICA A UNA E DUE DIMENSIONI	100.000	F
TP 681	ANALISI ABC CON LOTUS 1-2-3	100.000	F
TP 669	GESTIONE DELLE COMMESSE CON dBASE III PLUS	100.000	F
	MARKETING & MANAGEMENT		
M 648	PROBLEMI DI MARKETING	45.000	
M 649	DISTINTA BASE	23.000	
M 650	TECNICHE DI ANALISI FINANZIARIA	52.000	
M 647	RICERCHE DI MERCATO	72.000	
	NOVITÀ NOVEMBRE '88		
BE 737	IL MANUALE DEI FILTRI ATTIVI	34.000	
R 730	TURBO C	62.000	F
GY 662	UNIX - Architettura di sistema	65.000	
PP 672	MARKETING CON LOTUS 1 2 3	41.000	F
GY 703	LINGUAGGI DI PROGRAMMAZIONE	69.000	
BE 713	80286 HARDWARE	65.000	
M 679	DIRECT MARKETING	35.000	
M 706	TECNICHE DI MARKETING	43.000	
M 726	DIRECT MAILING	35.000	

GRUPPO EDITORIALE JACKSON

F = libro con floppy
C = libro con cassette

Per le vostre ordinazioni per corrispondenza utilizzate l'apposita cedola inserita in questa rivista.

Y0008/D

IN EDICOLA I PRIMI 2 FASCICOLI A SOLE 200 LIRE

NUOVA DA JACKSON

ENCICLOPEDIA PRATICA JACKSON DI

ELETTRICITÀ & ENERGIA

IMPIANTI E MATERIALI
FAI DA TE
BASI DELL'ELETTRICITÀ
ELETTRODOMESTICI
FONTI DI ENERGIA

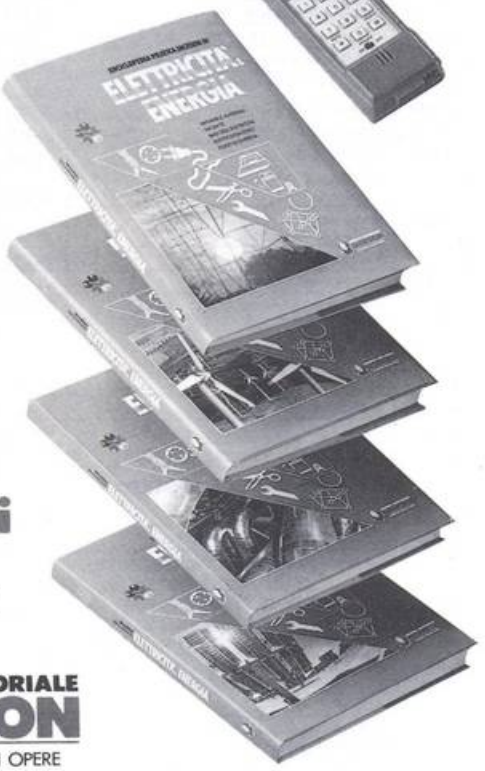
In Regalo
agli acquirenti dell'opera
un favoloso telefono a tastiera con memoria

ELETTRICITÀ & ENERGIA è la grande opera del Gruppo Editoriale Jackson nata per tutti coloro che intendono acquisire la padronanza più completa delle fonti energetiche, dalle tecnologie utilizzate, fino alle principali applicazioni.

Grande spazio è dedicato all'*elettricità*, dalle sue leggi fondamentali, fino ai suoi più comuni settori di utilizzo. L'elettricità è, infatti, tra tutte le risorse energetiche, quella, con cui chiunque di noi ha quotidianamente a che fare.

Rivolta all'hobbista oltre che al tecnico, ELETTRICITÀ & ENERGIA riserva un buon numero di pagine, in ogni fascicolo, anche a nozioni di tipo pratico, dall'impiantistica al "fai da te" elettrico.

Tutti gli argomenti sono trattati con lo stile e la professionalità delle Grandi Opere Jackson.



52 fascicoli
da rilegare in:
4 splendidi volumi
con un totale di 1050 pagine
oltre 5000 fotografie e illustrazioni

GRUPPO EDITORIALE JACKSON
DIVISIONE GRANDI OPERE

SERVIZIO LETTORI

IL GRUPPO EDITORIALE JACKSON PROMUOVE OGNI GIORNO NUOVE INIZIATIVE PER FACILITARE IL CONTINUO DIALOGO CON I PROPRI LETTORI. NATURALMENTE È IMPORTANTE CHE QUESTO SCAMBIO DI INFORMAZIONI SIA RESO IL PIÙ POSSIBILE AUTOMATICO E CHE I SUOI TEMPI SIANO SEMPRE PIÙ RISTRETTI. È CON QUESTO INTENTO CHE NASCE IL SERVIZIO LETTORI JACKSON, ORGANIZZATO IN MODO DA SODDISFARE OGNI ESIGENZA, SECONDO UN SISTEMA DI CEDOLE PRECONFIGURATE, DA INVIARE AL NOSTRO SERVIZIO MARKETING. ANZITUTTO, IL SERVIZIO LETTORI JACKSON CONSENTE DI ORDINARE, UTILIZZANDO LA CEDOLA DI COMMISSIONE LIBRERIA A FIANCO, I LIBRI DEL GRUPPO EDITORIALE JACKSON SCEGLIENDO LA MODALITÀ DI PAGAMENTO PREFERITA. UN ESTRATTO CONDENSATO DEL CATALOGO LIBRI E GRANDI OPERE JACKSON È PUBBLICATO NELLE ULTIME PAGINE DI QUESTA RIVISTA; IL CATALOGO COMPLETO PUÒ ESSERE

COMUNQUE ORDINATO, UTILIZZANDO LA CEDOLA NUMERO 3: INFORMAZIONI & AGGIORNAMENTI. QUEST'ULTIMA È LA PIÙ IMPORTANTE E PERMETTE AL LETTORE DI RICEVERE, DIRETTAMENTE A CASA PROPRIA, TUTTE LE INFORMAZIONI SULLE INIZIATIVE JACKSON CHE LO INTERESSANO: CATALOGHI, LIBRI, CAMPAGNA ABBONAMENTI CORSI DELLA DIVISIONE FORMAZIONE E ALTE TECNOLOGIE S.A.T.A., COPIE OMAGGIO DI RIVISTE E FASCICOLI DI GRANDI OPERE. QUESTO SERVIZIO CONSENTE, OLTRE CHE DI RIMANERE AGGIORNATI, ANCHE DI AGGIORNARE I COLLEGHI E GLI AMICI, POICHÈ LA CEDOLA È STUDIATA ANCHE CON QUESTO INTENTO. NON PIÙ TELEFONATE LETTERE: DA OGGI È SUFFICIENTE SPEDIRE L'APPOSITO TAGLIANDO, PER OTTENERE IN BREVISSIMO TEMPO IL MATERIALE DESIDERATO.

GRUPPO EDITORIALE JACKSON
PRIMO NELLA BUSINESS-TO-BUSINESS COMMUNICATION

CEDOLA COMMISSIONE LIBRI

Se desiderate ordinare libri Jackson, utilizzate questa cedola. Indicate negli appositi spazi i codici dei libri richiesti e le quantità. Precisate anche il tipo di pagamento scelto, il vostro nome, cognome, indirizzo. **Ritagliate e spedite, riportando sulla busta l'indirizzo esatto del Gruppo Editoriale Jackson.**

RITAGLIATE E SPEDITE IN BUSTA CHIUSA

MITTENTE

COGNOME _____

NOME _____

VIA E NUMERO _____

CAP _____ CITTÀ _____

PROV. _____ TEL. (_____) _____

GRUPPO EDITORIALE JACKSON
Via Rosellini, 12
20124 Milano

CEDOLA COMMISSIONE GRANDI OPERE

Se desiderate acquistare una enciclopedia o una "Grande Opera Jackson", con pagamento in un'unica soluzione oppure informazioni per l'acquisto con formula rateale a sole L. 25.000 mensili e un semplice anticipo di L. 45.000, compilate questa cedola precisando il tipo di pagamento scelto. **Ritagliate e spedite, riportando sulla busta l'indirizzo esatto del Gruppo Editoriale Jackson.**

RITAGLIATE E SPEDITE IN BUSTA CHIUSA

MITTENTE

COGNOME _____

NOME _____

VIA E NUMERO _____

CAP _____ CITTÀ _____

PROV. _____ TEL. (_____) _____

GRUPPO EDITORIALE JACKSON
Via Rosellini, 12
20124 Milano

CEDOLA INFORMAZIONI E AGGIORNAMENTI

Se desiderate ricevere rapidamente informazioni sui prodotti e attività del Gruppo Editoriale Jackson, barrate le caselle della cedola che vi interessano. **Ritagliate e spedite, riportando sulla busta l'indirizzo esatto del Gruppo Editoriale Jackson.**

RITAGLIATE E SPEDITE IN BUSTA CHIUSA

MITTENTE

COGNOME _____

NOME _____

VIA E NUMERO _____

CAP _____ CITTÀ _____

PROV. _____ TEL. (_____) _____

GRUPPO EDITORIALE JACKSON
Via Rosellini, 12
20124 Milano

SCUOLA DI ALTE TECNOLOGIE APPLICATE JACKSON S.A.T.A.

CALENDARIO CORSI 1988

Lunedì 29 agosto Inizio corso SPECIALIZZAZIONE IN AUTOMAZIONE INDUSTRIALE E ROBOTICA 175 ore (serale)
Inizio corso MICROPROCESSORI BASE 40 ore (serale)
Lunedì 5 settembre
Inizio corso CONTROLLO E PREVENZIONE DELLE PARTI ELETTRONICHE DALLE SCARICHE ELETTROSTATICHE 24 ore (intensivo)
Inizio corso PROGRAMMAZIONE IN BASIC 40 ore (intensivo)
Inizio corso UNIX, XENIX UTENTI 80 ore (serale)
Inizio corso DBIII PLUS UTENTI 24 ore (serale)
Lunedì 12 settembre
Inizio corso EMC-COMPATIBILITÀ ELETTROMAGNETICA 24 ore (intensivo)
Inizio corso PROGRAMMAZIONE IN PASCAL-TURBOPASCAL 50 ore (intensivo)
Inizio corso CASE-COMPUTER AIDED SOFTWARE ENGINEERING 40 ore (intensivo)
Giovedì 15 settembre
Inizio corso DBIII PLUS PROGRAMMAZIONE 24 ore (serale)
Lunedì 19 settembre
Inizio corso DESK TOP PUBLISHING 40 ore (serale)
Inizio corso TRASDUTTORI, SENSORI, ATTUATORI 20 ore (serale)
Mercoledì 21 settembre
Inizio corso ELETTRONICA DIGITALE 60 ore (intensivo)
Lunedì 26 settembre
Inizio corso PROGRAMMAZIONE IN COBOL 60 ore (intensivo)
Mercoledì 28 settembre
Inizio corso CONTROLLORI LOGICI PROGRAMMABILI 40 ore (serale)
Lunedì 3 ottobre Inizio corso ARCHITETTURA SNA 32 ore (intensivo)
Inizio corso INTEGRAZIONE EDP E TLC NELL'OFFICE AUTOMATION 32 ore (intensivo)
Inizio corso APPLICAZIONI INDUSTRIALI DELLE TECNICHE LASER DI BASSA POTENZA 32 ore (intensivo)
Inizio corso VENTURA 24 ore (intensivo)
Lunedì 10 ottobre
Inizio corso APPARATI E SISTEMI PER LE RETI DI COMPUTER 40 ore (intensivo)
Inizio corso MICROPROCESSORI BASE 40 ore (intensivo)
Inizio corso PAGE MAKER 24 ore (intensivo)
Lunedì 17 ottobre Inizio corso PROGRAMMAZIONE IN C 80 ore (intensivo)
Inizio corso PROGRAMMAZIONE WINDOWS BASE 80 ore (intensivo)
Inizio corso PROCESSORI DI SEGNALE DIGITALE 60 ore (intensivo)
Inizio corso MANUSCRIPT 24 ore (intensivo)
Lunedì 24 ottobre Inizio corso ELEMENTI BASE DI ROBOTICA 20 ore (serale)
Lunedì 7 novembre Inizio corso PC/MS-DOS 24 ore
Inizio corso PIANIFICAZIONE RETICOLARE COL PC 24 ore (intensivo)
Inizio corso OFFICE COMMUNICATION 24 ore (intensivo)
Inizio corso INFOCENTER 32 ore (intensivo)
Inizio corso TECNICHE BASE E SISTEMI PER TRASMISSIONI DATI 80 ore (serale)
Inizio corso MICROPROCESSORI EVOLUTO 40 ore (serale)
Inizio corso INTRODUZIONE ALL'INTELLIGENZA ARTIFICIALE E AI SISTEMI ESPERTI 40 ore (intensivo)
Inizio corso INFORMIX/SQL 50 ore (serale)
Inizio corso ARCHITETTURA OS/2 40 ore (intensivo)
Inizio corso MICROPROCESSORI A 16 BIT 60 ore (intensivo)
Inizio corso AFFIDABILITÀ DEI CIRCUITI E DEI COMPONENTI ELETTRONICI 24 ore (intensivo)
Mercoledì 9 novembre Inizio corso WORD 24 ore
Lunedì 14 novembre
Inizio corso RETI DI COMUNICAZIONE NELLA FABBRICA AUTOMATIZZATA 20 ore (serale)
Inizio corso MODELLI PREVISIONALI COL PC 24 ore (intensivo)
Inizio corso USO DEL PC NELL'AREA PRODUZIONE 24 ore (intensivo)
Inizio corso PROGETTAZIONE DEI MODERNI CIRCUITI STAMPATI 24 ore (intensivo)
Lunedì 21 novembre
Inizio corso RETI A COMMUTAZIONE DI PACCHETTO 40 ore (intensivo)
Inizio corso PROGRAMMAZIONE IN LISP 40 ore (intensivo)
Inizio corso MODELLI DECISIONALI COL PC 24 ore (intensivo)
Inizio corso USO DEL PC NELL'AREA MARKETING 24 ore (intensivo)
Lunedì 28 novembre Inizio corso MULTIPLAN 24 ore
Inizio corso PROGRAMMAZIONE IN PROLOG 40 ore (intensivo)
Inizio corso AUTO-CAD 32 ore (serale)
Inizio corso SERVIZI A VALORE AGGIUNTO SULLE RETI X25 24 ore (intensivo)
Mercoledì 30 novembre Inizio corso LOTUS 1-2-3 24 ore
Lunedì 12 dicembre
Seminario con WORK-SHOP SUI LINGUAGGI DELLA IV GENERAZIONE 24 ore (intensivo)
Inizio corso IL MODELLO OSI 32 ore (intensivo)
Inizio corso SYMPHONY 40 ore (intensivo)
Inizio corso OTTIMIZZAZIONE E DEBUGGING "C" 40 ore (intensivo)
Inizio corso MICROPROCESSORI EVOLUTO 40 ore (intensivo)

SCUOLA DI ALTE TECNOLOGIE APPLICATE

Per le modalità di iscrizione e richiesta di programmi dettagliati, telefonare alla DIVISIONE FORMAZIONE E PRODOTTI PER LA DIDATTICA del Gruppo Editoriale Jackson
Via Imperia 2 Milano
Telefono 8467500

S.A.T.A.

SÌ INVIATEMI I VOLUMI SOTTOELENCATI:

INDICARE CHIARAMENTE CODICE, QUANTITÀ E PREZZO DEI VOLUMI RICHIESTI

Codice	Quantità	Prezzo		Codice	Quantità	Prezzo

Ordine minimo L. 30.000 + L. 3.500 per contributo fisso spese di spedizione

☐ **Non sono abbonato** a riviste Jackson

☐ **Sono abbonato** alla/e seguente/i rivista/e Jackson _____

_____ e ho quindi diritto allo sconto del **10%**

MODALITÀ DI PAGAMENTO

☐ Allego assegno n. _____ di L. _____

della Banca _____

☐ Ho effettuato il pagamento di L. _____ a mezzo:
☐ vaglia postale ☐ vaglia telegrafico ☐ versamento sul c/c postale n. 11666203 intestato a Gruppo Editoriale Jackson SpA Milano e allego fotocopia della ricevuta.

☐ Pagherò al postino l'importo di L. _____ al ricevimento dell'opera.

☐ Vi autorizzo ad addebitare l'importo di L. _____
sulla carta di credito: ☐ Visa ☐ American Express ☐ Diners Club ☐ Carta SI

conto n. _____ data di scadenza _____

☐ Richiedo l'emissione della fattura (formula riservata alle aziende) e comunico il numero di Partita IVA _____

DATA _____ FIRMA _____

SI INVIATEMI LE SEGUENTI "GRANDI OPERE JACKSON"

☐ **EI - Enciclopedia di Elettronica e Informatica** 10 volumi	L. 476.000
☐ **SOFTWARE** 5 volumi	L. 236.000
☐ **DEI - Dizionario di Elettronica e Informatica** 10 volumi	L. 276.000
☐ **Enciclopedia Monografica di Elettronica e Informatica** 2 volumi	L. 116.000
☐ **ABC Personal Computer** 4 volumi	L. 136.000
☐ **VIDEOBASIC MSX** 20 lezioni + 20 cassette	L. 176.000
☐ **VIDEOBASIC SPECTRUM** 20 lezioni + 20 cassette	L. 176.000
☐ **VIDEOBASIC** C64/C128/64PC 20 lezioni + 20 cassette	L. 176.000
☐ **VIDEOBASIC** C64/C128/64PC 20 lezioni + 10 floppy	L. 176.000
☐ **VIDEOBASIC** C16/PUS4 20 lezioni + 20 cassette	L. 176.000
☐ **VIDEOBASIC** VIC20 20 lezioni + 20 cassette	L. 176.000
☐ **CORSO DI GRAFICA** C64/C128/64PC 10 lez. + 10 cass.	L. 96.000
☐ **A SCUOLA DI SCACCHI** C64/C128/64PC 10 lezioni + 10 cassette	L. 96.000
☐ **7 NOTE BIT** C64/C128/64PC 15 lezioni + 15 cassette	L. 156.000
☐ **LABORATORIO DI ELETTRONICA** 5 volumi (disp. da giugno 1988)	L. 236.000
☐ **BYTES** 6 volumi (disp. da giugno 1988)	L. 276.000

☐ **PAGAMENTO RATEALE**

MODALITÀ DI PAGAMENTO IN UN'UNICA SOLUZIONE

☐ Allego assegno n. _____ di L. _____

della Banca _____

☐ Ho effettuato il pagamento di L. _____ a mezzo:
☐ vaglia postale ☐ vaglia telegrafico ☐ versamento sul c/c postale n. 11666203 intestato a Gruppo Editoriale Jackson SpA Milano e allego fotocopia della ricevuta.

☐ Pagherò al postino l'importo di L. _____ al ricevimento dell'opera.

☐ Vi autorizzo ad addebitare l'importo di L. _____
sulla carta di credito: ☐ Visa ☐ American Express ☐ Diners Club ☐ Carta SI

conto n. _____ data di scadenza _____

☐ Richiedo l'emissione della fattura (formula riservata alle aziende) e comunico il numero di Partita IVA _____

DATA _____ FIRMA _____

CEDOLA INFORMAZIONI E AGGIORNAMENTI

☐ Desidero riceve informazioni sulle **riviste** JACKSON
☐ Desidero ricevere una copia saggio della **rivista** JACKSON _____
(indicare titolo)
☐ Desidero ricevere il **catalogo libri tecnici** JACKSON
☐ Desidero ricevere il **catalogo libri scolastici** JACKSON
☐ Desidero ricevere il **catalogo grandi opere** JACKSON e una copia saggio _____
(indicare titolo)
☐ Desidero ricevere informazioni per l'acquisto in **forma rateale delle Grandi Opere** JACKSON e una copia saggio _____
(indicare titolo)
☐ Desidero ricevere il **catalogo occasioni** JACKSON

Desidero ricevere maggiori informazioni sulla **Divisione Formazione e Prodotti per la Didattica**
☐ Area ELETTRONICA E MICROPROCESSORI
☐ Area TELECOMUNICAZIONI E TELEMATICA
☐ Area AUTOMAZIONE INDUSTRIALE E ROBOTICA
☐ Area INFORMATICA E INTELLIGENZA ARTIFICIALE
☐ Area ALTE TECNOLOGIE SPECIALI
☐ Laboratorio di INFORMATICA
☐ Laboratorio di TELEMATICA
☐ Laboratorio di OFFICE AUTOMATION
☐ Laboratorio di ELETTRONICA
☐ Laboratorio di MICROPROCESSORI
☐ Laboratorio di AUTOMAZIONE INDUSTRIALE

LA TUA COLLANA PER CAPIRE, IDEARE, PROGETTARE

LIBRI DI BASE
ELETTRONICA

ELETTRONICA IN AUTO

L'elettronica mantiene uno stretto rapporto con l'automobile, sin dall'istante della progettazione. In questa guida la presentazione dei circuiti elettronici in dotazione sulle vetture in circolazione e su quelle che lo saranno nel prossimo futuro. Un argomento di grande interesse e di estrema utilità.

AMPLIFICATORI OPERAZIONALI

Gli OP-AMP per la loro caratteristica di poter "eseguire operazioni", sono oggi impiegati e applicati in diversi settori dell'elettronica, nell'audio e alta fedeltà, nelle apparecchiature di laboratorio, nell'elettronica di commutazione. Questa guida ti permetterà di familiarizzare con questo componente elettronico moderno.

Se hai l'esigenza di conoscere per costruire tutto sull'elettronica, il Gruppo Editoriale Jackson ti propone i nuovi: "Libri di Base Elettronica", 20 preziose guide attraverso circuiti, componenti, grafici, fotografie e soprattutto innumerevoli idee per scatenare la tua fantasia con progetti collaudati e di immediata realizzazione.

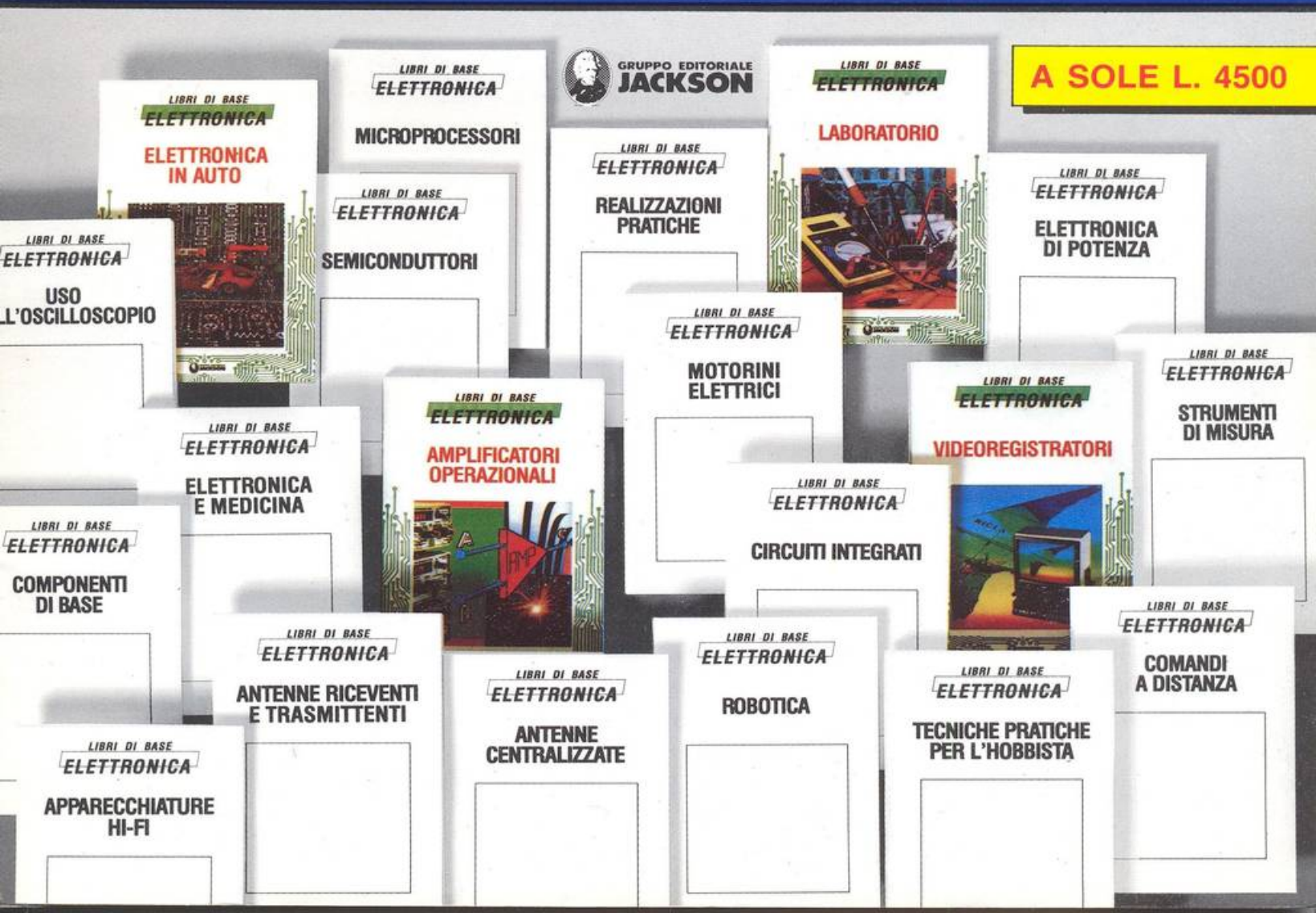
VIDEOREGISTRATORI

In questo libro viene fornita un'idea di base sui differenti tipi di magnetoscopi esistenti in commercio, con particolare riguardo ai sistemi domestici da mezzo pollice e ai sistemi industriali U-Matic. Un intero capitolo è dedicato al montaggio, con consigli sulla manutenzione e sulla conservazione delle attrezzature.

LABORATORIO

Sei un appassionato di laboratorio, oppure miri a diventare un professionista del settore? Ecco un testo in grado di chiarirti le idee, verificare i tuoi progetti o insegnarti a regolare le apparecchiature già costruite. Un tester ed un saldatore per iniziare e poi la tua fantasia...

QUESTO MESE IN EDICOLA



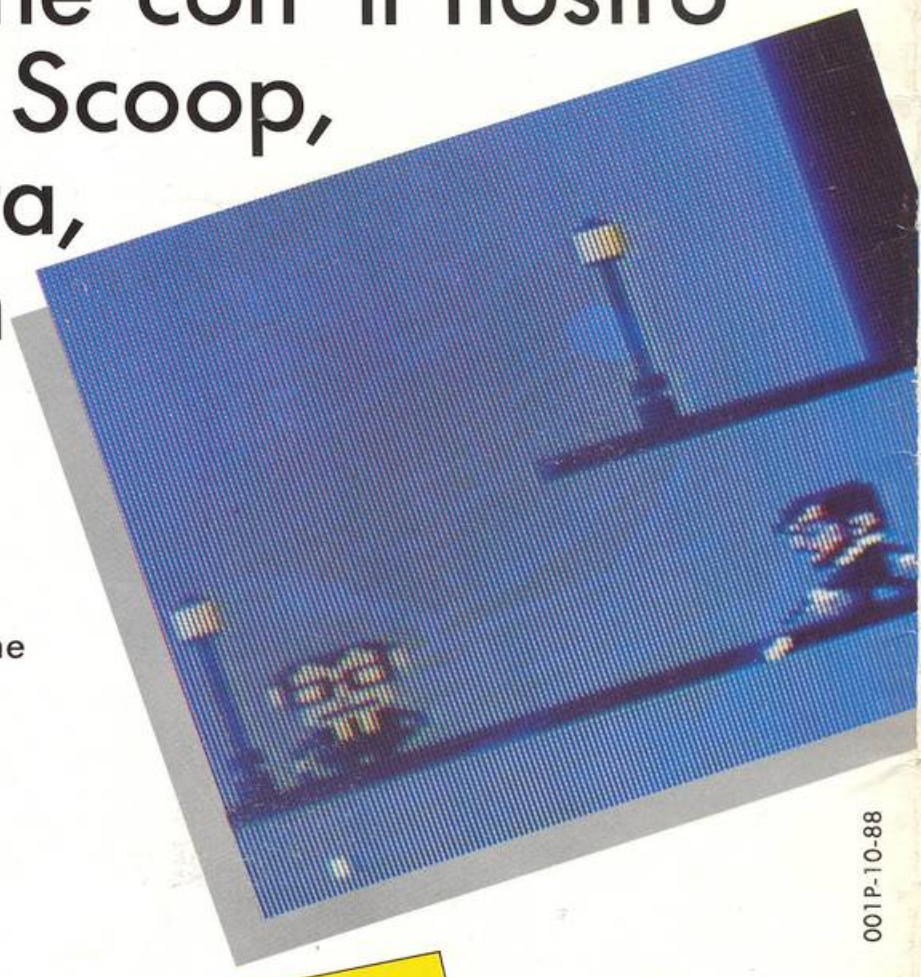
GRUPPO EDITORIALE JACKSON

A SOLE L. 4500

IN EDICOLA CON SUPER COMMODORE 64 e 128 N. 21

SATYRIGAME

Un fantastico videogioco inedito. Thriller e azione con il nostro eroe Eugenio Scoop, noto giornalista, alle prese con agguerriti parlamentari.



SATYRIGAME è, come lo stesso nome indica, un gioco satirico d'azione che si sviluppa in ambiente parlamentare e ha como scopo, da parte del protagonista (giornalista di grande fama), quello di raggiungere indenne la redazione del suo giornale per pubblicare una notizia bomba... non perdetevi il resto.

001P-10-88

GRUPPO EDITORIALE JACKSON

E IN PIÙ IL FAVOLOSO SUPERPOSTER "TROLL" IN REGALO!